

Frühjahrssemester 2012

Semesterarbeit LEM 1201

Automatischer Teststand für solarmodulintegrierte DC-DC-Konverter

Axel Bomhauer-Beins



Betreuer: M.Sc. Matthias Kasper
Professor: Prof. Dr. Johann W. Kolar

Aufgabenstellung

Semesterarbeit FS 2012

Axel Bomhauer-Beins

Vollautomatischer Teststand für solar-modulintegrierte DC-DC-Konverter

A Beschreibung



Erneuerbaren Energien, insbesondere der Photovoltaik (PV), kommt eine stetig wachsende Bedeutung bei der Stromerzeugung zu. Heutige PV-Anlagen bestehen aus einer einfachen Serienschaltung von Solarmodulen. Aufgrund dieser Verschaltung geht die Leistung von lokal abgeschatteten Modulen (z. B. durch lokale Verschmutzung, Schatten von Schornsteinen etc.) verloren. Mit dem Einsatz von Konvertern auf Solarmodulebene kann erreicht werden, dass jedes Modul seine maximale momentane Leistung ins Stromnetz einspeisen kann.

Da eine hohe Effizienz der Konverter für eine lange Lebensdauer und große Energieausbeute wichtig ist, soll in dieser Arbeit ein automatisiertes Messsystem entwickelt werden, mit dem die Effizienz der auf dem Markt erhältlichen Konverter und von eigenen Prototypen gemessen werden kann. Zur Gesamteffizienz eines solarmodulintegrierten DC-DC-Konverters zählt neben der Effizienz der Leistungselektronik auch die Genauigkeit des durchgeführten Maximum Power Point (MPP) Trackings.

Zur Messung werden hochgenaue Messgeräte zentral über einen PC per LABVIEW angesteuert und ausgelesen. Die Datenaufbereitung erfolgt danach in MATLAB. Dieser Vorgang soll automatisiert und über ein Interface durchgeführt werden, mit direkter Anzeige der berechneten Effizienz in verschiedenen Arbeitspunkten.

Diese Arbeit ist Teil eines übergeordneten Forschungsprojekts mit dem Ziel, einen hocheffizienten solarmodulintegrierten DC-DC-Konverter zu entwickeln. Bei dieser Arbeit können vertiefte Kenntnisse über die Leistungselektronik und Anforderungen im Bereich der Photovoltaik gewonnen werden.

B Aufgabenstellung

Im Rahmen dieser Semesterarbeit soll die Programmierung der Messgeräte-Ansteuerung erfolgen. Folgende Arbeitsschritte sind nötig:

- Einarbeiten in die Anforderungen an PV DC-DC-Konverter
- Aufstellen eines Zeitplans
- Recherche über kommerziell verfügbare PV DC-DC-Konverter und Bestellung
- Literaturstudium über hochgenaue (Effizienz)messungen
- Einarbeitung in LABVIEW und die Bedienung der Testhardware
- Programmierung der Ansteuerung der Messgeräte in LABVIEW mit GUI
- Programmierung der Datenauswertung in MATLAB:
  - Effizienz der Konverter in dem gemessenen Arbeitspunkt
  - MPP Tracking-Genauigkeit
- Durchführung von Messungen mit bestellten Konvertern in verschiedenen Arbeitspunkten (Volllast und Teillastbereiche)
- Optional: Durchführung der Messung bei verschiedenen Temperaturen
- Ausführliche Dokumentation der durchgeführten Arbeiten
- Schlusspräsentation der Semesterarbeit

C Allgemeine Hinweise

Die beiliegenden „Vorschriften über die Durchführung von Studien- und Diplomarbeiten“ (gelbes Merkblatt) sind Bestandteil dieser Aufgabenstellung. Die „Sicherheitsvorschriften“ (rotes Merkblatt) müssen strikte eingehalten werden. Das ausgefüllte, weiße Formular über die „Gefahrenabschätzung am Arbeitsplatz“ sowie die „Handhabung der Software“ ist bei Beginn der Arbeit Herrn Peter Albrecht (ETL H11) abzugeben. Die Abgabekarte ist bis zum Schluss der Studienarbeit aufzubewahren.

Termine:	Arbeitsbeginn:	20.02.2012
	Arbeitsende:	01.06.2012
	Schlusspräsentation:	Juni 2012, Termin nach Vereinbarung
	Abgabe der Arbeit:	Juni 2012, Termin nach Vereinbarung

Betreuung: Matthias Kasper, ETL I14
044 632 61 14, kasper@lem.ee.ethz.ch

Beilagen: 2 Merkblätter, 1 Formular
1 Kopierkarte, 1 Abgabekarte

Prof. Dr. J. W. Kolar

Inhaltsverzeichnis

Aufgabenstellung **III**

Vorwort **VII**

Notation **IX**

Zusammenfassung **XIII**

1 Einführung **1**

2 Nutzung der Photovoltaik **3**
2.1 Prinzip photoelektrischer Energieerzeugung . . . 3
2.2 Verschaltung von Solarmodulen . . . 5
2.2.1 Serienschaltung mit nachgestelltem Wechselrichter . . . 5
2.2.2 Direkte Netzeinspeisung . . . 7
2.2.3 Parallelschaltung . . . 8
2.3 MPP-Tracking . . . 8

3 Aufbau des Teststandes **11**
3.1 Versuchsanordnung . . . 11
3.2 Messgeräte . . . 12
3.2.1 Überblick . . . 12
3.2.2 Solar Array Simulator „Agilent E4360A“ . . . 12
3.2.3 Tischmultimeter „Agilent 34410A“ . . . 15
3.2.4 burster Hochpräzisionswiderstand, Modell 1282 . . . 17
3.2.5 Funktionsgenerator „Agilent 33220A“ . . . 17
3.2.6 Elektronische Last „Chroma 63202 2,6 kW“ . . . 17
3.3 Prüflinge . . . 18
3.4 Berechnung der Ergebnisse . . . 18
3.5 Messgenauigkeit . . . 19
3.5.1 Einführung . . . 19
3.5.2 Zeitpunkt der Messung . . . 19
3.5.3 Unsicherheiten der Geräte . . . 20
3.5.4 Ungenauigkeiten durch Aufbau und Verkabelung . . . 21

4 Software **23**
4.1 Anforderungen . . . 23
4.2 Konzept . . . 23
4.3 Grundlagen der Programmierung . . . 24
4.4 Funktionalität . . . 25
4.4.1 Einführung . . . 25

4.4.2 Start der Oberfläche . . . 25
4.4.3 Auswahl der Schaltung . . . 26
4.4.4 Starten, Herunterfahren und Neustarten des Systems . . . 26
4.4.5 Steuerung des Solar Array Simulators . . . 26
4.4.6 Anzeige der Solarkurven . . . 27
4.4.7 Steuerung der Last . . . 27
4.4.8 LiveView . . . 29
4.4.9 Durchführung von Messungen . . . 29
4.4.10 Auswerten und Speichern von Messungen . . . 31
4.4.11 Abbruch einer Messung . . . 32
4.4.12 Systemweite Funktionen in der Hauptkontrolle . . . 32
4.4.13 Anzeige der Ergebnisse . . . 33

5 Ergebnisse der Testmessungen **35**
5.1 Allgemeine Feststellungen . . . 35
5.2 Testmessungen am Konverter SolarMagic SM3320-1A1 . . . 36

6 Ausblick **43**

Literaturverzeichnis **45**

Abbildungsverzeichnis **47**

Tabellenverzeichnis **49**

Anhang **52**

A Software-Dokumentation **53**
A.1 Überblick . . . 53
A.2 Grafische Oberflächen . . . 54
A.3 Globale Variablen . . . 56
A.4 Funktionen und Skripte . . . 60

B Vorgehen beim Austausch eines Gerätes **83**

C Technische Daten der Messgeräte **85**
C.1 Solar Array Simulator . . . 85
C.2 Tischmultimeter . . . 85
C.3 burster Hochpräzisionswiderstand . . . 86
C.4 Elektronische Last . . . 86

Eigenständigkeitserklärung **87**

Vorwort

Die Diskussion um Energie ist allgegenwärtig: Ausstieg aus der Atomkraft? Wie lange reichen die Ölreserven noch? Gibt es Möglichkeiten, elektrische Energie „sauber“, also ohne Schaden für die Umwelt, zu generieren?

Im Fokus all dieser Diskussionen stehen im Endeffekt die erneuerbaren Energien – Wasserkraft, Windkraft, Solarenergie. Sie gelten teilweise – trotz ihrer durchaus vorhandenen Nachteile – als Allheilmittel für unsere Energieprobleme. Eine Aufgabe der Forschung ist es also, diese Nachteile der erneuerbaren Energie zu erkennen und Lösungen zu finden.

Im Zentrum dieser Arbeit steht die Solarenergie: Um elektrische Energie aus Solarzellen in das bestehende elektrische Netz einspeisen zu können, müssen mehrere Solarzellen zu einem Solarmodul zusammengeschaltet werden, anschließend müssen wieder mehrere Solarmodule zusammengeschaltet werden. Geschieht dies ohne weitere Maßnahmen, so können verschattete Solarmodule thermisch zerstört werden und so unter Umständen einen Hausbrand auslösen. Um diesem Effekt entgegenzuwirken, können mittels Dioden jene Module abgeschaltet werden, denen die thermische Zerstörung droht – was jedoch auch zum Verlust der theoretisch noch gewinnbaren Leistung führt.

In den vergangen Monaten habe ich mich damit befasst, zur Untersuchung eines neueren Lösungsansatzes beizutragen: Mit Hilfe von Gleichspannungswandlern ist es möglich, aus jedem Solarmodul die maximale Leistung herauszuholen. Dies ist jedoch nur dann praktikabel, wenn die Konverter so effizient sind, dass ihr Einsatz auch wirklich (ökonomisch) sinnvoll ist. Um dies zu überprüfen und der künftigen Entwicklung ein Testwerkzeug zur Verfügung zu stellen, habe ich einen Teststand entwickelt, der genau diese Konvertereffizienz unter verschiedenen Bedingungen testet. Mit dieser Arbeit möchte ich einen kleinen Beitrag zur Weiterentwicklung der erneuerbaren Energien leisten und dem geneigten Leser einen Eindruck vom faszinierenden Thema der Photovoltaik und der Leistungselektronik geben.

An dieser Stelle möchte ich auch meinen Dank aussprechen:

- Meinem Betreuer Matthias Kasper für seine Unterstützung und die hervorragende Betreuung während der ganzen Arbeit – und ganz besonders dafür, dass er durch seine Anregungen, Rückfragen und Vorschläge den Arbeitsprozess stets bereichert und angeregt hat,
- der Professur für Leistungselektronik und Messtechnik der ETH Zürich für das Zurverfügungstellen des Laborplatzes, der Hardware für den Messaufbau und die übrige Infrasturktur,
- Herrn Peter Albrecht für das Beschaffen der Evaluations-Kits der Konverter und das Bereithalten von elektrischen und elektronischen Bauteilen sowie
- allen anderen, die mich in irgendeiner Form während dieser Arbeit unterstützt haben.

Zürich, im Mai 2012 Axel Bomhauer-Beins

Notation

Im Folgenden sind Abkürzungen und Symbole aufgelistet, die in dieser Arbeit verwendet werden. Ebenso ist ihre Bedeutung angegeben, sofern im Text nicht explizit eine andere Definition gegeben wird.

Abkürzungen

AC	Alternating Current, dt. Wechselstrom
BNC	Bayonet Neill Concelman; eine koaxiale Steckverbindung
DC	Direct Current, dt. Gleichstrom
GPIB	General Purpose Interface Bus
MPP	Maximum Power Point; Punkt größtmöglicher Leistung
NPLC	Number of Power Cycles, dt. Anzahl Netzzyklen
SAS	Solar Array Simulator
SCPI	Standard Commands for Programmable Instruments
USB	Universal Serial Bus

Symbole und Einheiten

α	Temperaturkoeffizient
Δ	Symbol für Differenz
η	Effizienz
A	Einheit Ampère; implizite Annahme: Gleichstrom
c	Lichtgeschwindigkeit
e	Fehler
f_n	Netzfrequenz; $f_\mathrm{n} = 50$ Hz
I	Zeitunabhängiger Strom
l	Länge

P	Leistung
P_V	Verlustleistung
ppm	parts per million: Teile pro eine Million Teile
S	Einheit Siemens (Leitwert)
t	Zeit
T	Temperatur
V	Einheit Volt; implizite Annahme: Gleichspannung
V∼	Einheit Volt für Wechselspannungen
V	Zeitunabhängige Spannung
W	Einheit Watt
$\overline{x}$	Mittelwert der Größe x

Indizes

abs	tiefgestellt	Absoluter Wert, z. B. Fehler
c	tiefgestellt	Größe eines Konverters
def	tiefgestellt	Definierter Wert
eff	tiefgestellt	Effektivwert
in	tiefgestellt	Wert an einem Eingang
int	tiefgestellt	Integration
L	tiefgestellt	(Elektronische) Last
max	tiefgestellt	Größter zulässiger Wert
mess	tiefgestellt	Messwert oder auf einen Messwert bezogen
min	tiefgestellt	Kleinster zulässiger Wert
mpp	tiefgestellt	Wert am Punkt größtmöglicher Leistung
nom	tiefgestellt	Nominalwert
oc	tiefgestellt	Wert im Leerlauf (open circuit)
out	tiefgestellt	Wert an einem Ausgang
P	tiefgestellt	Parallelschaltung
PV	tiefgestellt	Größe eines Solar*moduls*
R	tiefgestellt	Wert eines Widerstandes
real	tiefgestellt	Realer Wert

rel	tiefgestellt	Relativer Wert, z. B. Fehler
th	tiefgestellt	Thermische Größe
TMM	tiefgestellt	Wert eines Tischmultimeters
tot	tiefgestellt	total
typ	tiefgestellt	Typischer Wert
S	tiefgestellt	Serienschaltung
sc	tiefgestellt	Wert im Kurzschlussfall (short circuit)

Zusammenfassung

Um Solarmodule effizienter nutzen zu können, wurden Gleichspannungswandler entwickelt, die von jedem Solarmodul die maximal mögliche Leistung ins Netz einspeisen sollen. Dabei ist es selbstverständlich von großem Interesse, diese Konverter, die direkt in die Solarmodule integriert werden sollen, mit möglichst hoher Effizienz zu bauen.

Zum aktuellen Zeitpunkt sind bereits verschiedene Konvertertopologien von verschiedenen Herstellern kommerziell verfügbar. Um diese Konverter auf ihre Effizienz hin zu prüfen, wird in dieser Arbeit ein Teststand entwickelt, der sowohl für eine serielle als auch eine parallele Verschaltung der Konverter geeignet ist – auch andere Gleichspannungswandler, die nichts mit Photovoltaik zu tun haben, können mit diesem Teststand geprüft werden.

Nachdem das Prinzip der photovoltaischen Energieerzeugung in Grundzügen verstanden wurde, ergaben Recherchen verschiedene (kommerziell verfügbare) Ansätze, um Solarmodule so effizient wie möglich zu nutzen. Von größtem Interesse sind dabei der Einsatz von Gleichspannungswandlern am Solarmodul mit anschließender Serienschaltung der Konverterausgänge wie auch der Einsatz von Gleichspannungswandlern mit anschließender Parallelschaltung der Konverterausgänge.

Die Beschaffung solcher Gleichspannungswandler ohne die gleichzeitige Bestellung einer Photovoltaik-Anlage gestaltet sich sehr schwierig. Lediglich Evaluations-Boards eines Konverters konnten einfach beschafft werden, einige Zeit später solche eines anderen Konverters. Es wurde nur ein Anbieter in den USA ausfindig gemacht, der mehrere der ermittelten Gleichspannungswandler im Angebot hat – die Lieferung wäre demzufolge mit viel Aufwand und hohen Transportkosten verbunden gewesen.

Der Teststand selbst wurde aus hochpräzisen Messgeräten aufgebaut und wird von einem PC mittels MATLAB gesteuert. Alle Geräte können – je nach gewünschtem Test – konfiguriert werden; es werden Einzelmessungen zur Untersuchung eines Arbeitspunktes sowie Messreihen zur Untersuchung des Verhaltens bei verschiedenen Parametern unterstützt. Der *LiveView*-Modus ermöglicht es, einen ungefähren Überblick über den aktuellen Zustand der Schaltung zu erhalten.

Je nach Schaltungsart der Konverterausgänge – seriell oder parallel – müssen einige Verbindungen am Teststand umgesteckt werden; die Software unterstützt diese Umstellung mittels Schaltfläche.

Nach der Konfiguration der Messung erfolgt diese vollständig automatisiert. Die Messwerte werden ausgelesen und die benötigten und gewünschten Größen der Schaltung (wie z. B. die Effizienzen) sowie ihre Messunsicherheit berechnet. Die Auswertung erfolgt in Form von Graphen, die die Messergebnisse veranschaulichen und Interpretationen ermöglichen. Für Einzelmessungen werden auch die gemessenen Werte in einem Schaltplan, der Teil der Benutzeroberfläche ist, dargestellt.

Die Graphen wie auch die Werte (die in einer globalen Variablen abgelegt sind) können gesichert werden – entweder durch das Speichern der MATLAB-Figur in einem beliebigen Format und / oder durch das Speichern der Ergebnis-Variablen in eine mat-Datei.

Weiterhin wurden mit dem Teststand einige Testmessungen durchgeführt, die bereits erste Aussagen über das Verhalten der getesteten Gleichspannungswandler ermöglichen. So konnte beispielsweise festgestellt werden, dass die Konverter eine gewisse Zeit benötigen, um wirklich die maximale Leistung aus dem Solarmodul ins Netz oder den Zwischenkreis einspeisen zu können.

1 Einführung

Wir leben in einer Welt, in der die elektrische Energie immer mehr an Bedeutung gewinnt: Nebst den Bemühungen der Transportindustrie – insbesondere im Automobilbereich – verstärkt elektrische Energie einzusetzen, verzeichnet die elektrische Energie auch einen steigenden Anteil am sogenannten „Endenergieverbrauch“[1]. In der Schweiz ist ihr Anteil von 2000 bis 2010 von 22% auf 23,6% angewachsen [BFE 2000, BFE 2010, Fig. 2], der „Gesamtverbrauch“ stieg im selben Zeitraum von knapp 54 000 GWh auf fast 60 000 GWh pro Jahr [BFE 2010, Tab. 25]. Der Photovoltaik fällt dabei eine tragende Rolle zu: Während im Jahr 2000 noch 1 325 Photovoltaik-Anlagen mit einer Nennleistung von insgesamt 15,2 MW installiert waren, so waren es 2010 bereits 7 425 Anlagen mit einer Gesamt-Nennleistung von 110,9 MW [BFE 2010, Tab. 32] – eine Steigerung um mehr als einen Faktor 7 in der Nennleistung![2]

Insbesondere auf Grund des steigenden Bedarfs an elektrischer Energie, nicht zuletzt aber auch aus wirtschaftlichen Gründen, besteht Interesse daran, die Photovoltaik so effizient wie möglich zu nutzen. Um dieses Ziel zu erreichen, gibt es bereits verschiedene Ansätze, auf die in Abschnitt 2.2 eingegangen wird. Einer dieser Ansätze ist es, die Verschaltung von Solarmodulen anzupassen und jedes Modul respektive jeden String von Modulen mit einem eigenen Konverter zu versehen und anschließend die Ausgänge der Konverter seriell oder parallel an einen Wechselrichter anzuschließen. So soll auch aus lokal abgeschatten Modulen[3] ein Maximum an Leistung gewonnen werden, anstatt diese Module abzuschalten. Eine hohe Konvertereffizienz ist dabei einerseits zwecks einer möglichst hohen Energieausbeute, andererseits für eine möglichst lange Lebensdauer der Konverter wünschenswert.[4]

Im Rahmen dieser Arbeit soll zur Überprüfung der Effizienz insbesondere dieser Module, aber auch des gesamten Ansatzes, ein automatischer Teststand entwickelt werden. Mit diesem Teststand soll es möglich sein, die Effizienz jeglicher Photovoltaik-Konverter zu überprüfen.

Um diesen Teststand entwickeln zu können, wird zuerst eine kurze Einführung in das Prinzip der Photovoltaik und die möglichen Verschaltungen und Ansätze gegeben (Kapitel 2); in den Kapiteln 3 und 4 wird der Aufbau des Teststandes (Hardware und Software) beschrieben. Die Ergebnisse einiger Testmessungen werden in Kapitel 5 dargestellt.

[1] „Endenergie beinhaltet [...] die vom Konsumenten für einen bestimmten Nutzen eingekaufte bzw. selbst produzierte Energie, wie zum Beispiel Strom für Licht oder Benzin fürs Auto.“ [BFE 2010, S. ii]

[2] Diese Zahlen beziehen sich auf die Erzeugung von elektrischer Energie; Anlagen zur direkten Beheizung oder mit anderen Aufgaben sind *nicht* berücksichtigt.

[3] Unter *lokal abgeschatteten Modulen* werden *Teile* einer Photovoltaik-Anlage verstanden, die durch *kleine, lokale* Schatten – wie z. B. den eines Schornsteins auf einem Dach – verdunkelt sind.

[4] Verluste im Konverter führen zu Erwärmung, welche die Alterung der Bauteile fördert und somit die Lebensdauer reduziert.

2 Nutzung der Photovoltaik

2.1 Prinzip photoelektrischer Energieerzeugung

Solarzellen, aus denen Photovoltaik-Anlagen aufgebaut sind, basieren auf den Prizipien der Halbleiterphysik.

Grundsätzlich sind Solarzellen klassische p-n-Übergänge. Treffen Photonen auf den Übergang, so können sie – bei korrekter Abstimmung der Halbleitermaterialien auf die Photonenwellenlänge – Elektronen aus dem Valenzband „herausschlagen“ und ins Leitungsband befördern. Die freien Elektronen wandern in Richtung des geringeren Energieniveaus; die entstandenen Löcher wandern in Richtung des höheren Energieniveaus (siehe **Abbildung 2.1**).

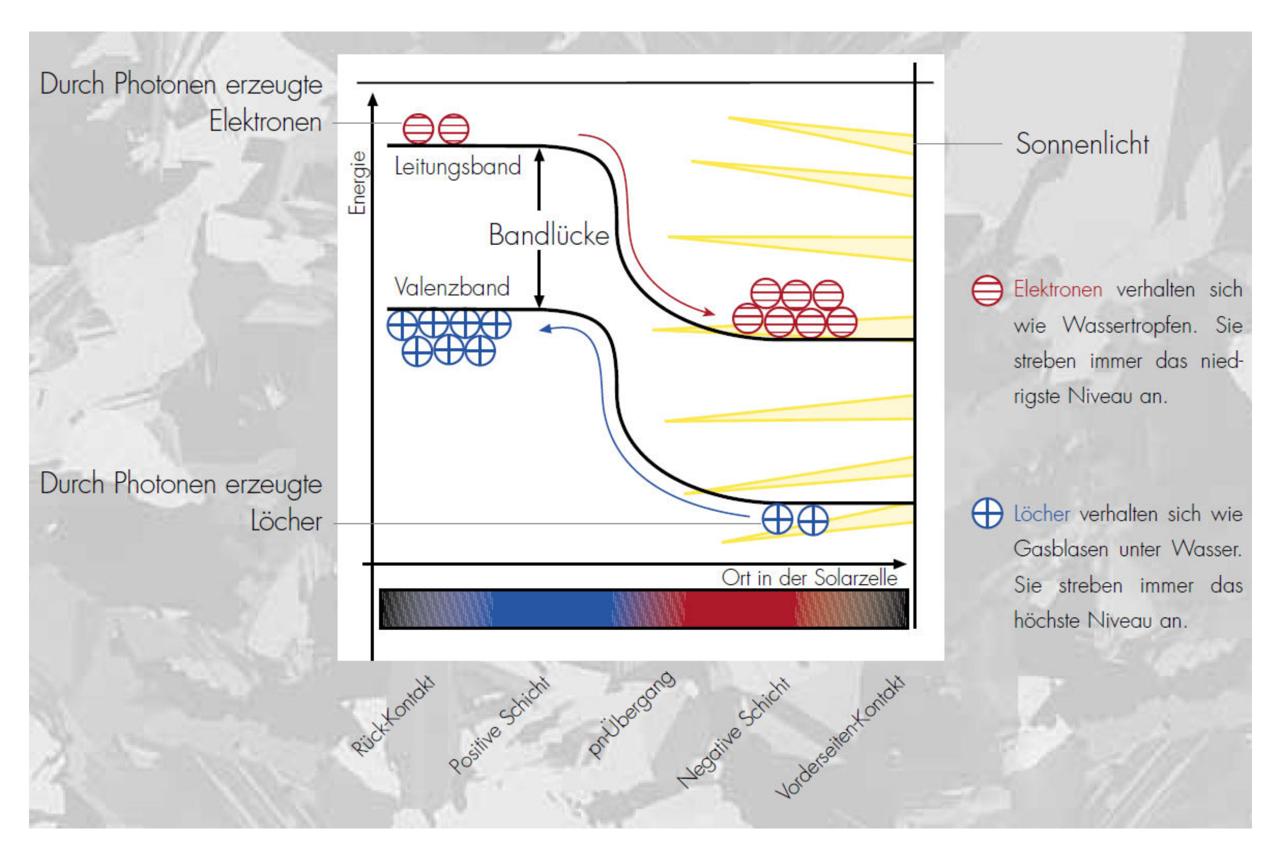


Abbildung 2.1: *Schematisches Banddiagramm eines p-n-Übergangs. Aus [DSolar o.J., Abb. 2]*

So entsteht zwischen den beiden Enden der Solarzelle, an denen sich die Klemmen befinden, eine Spannung von 0,5 V bis 0,7 V [Glotzbach 2009, S. 15].

Durch die Verschaltung mehrerer Solarzellen zu einem Modul entsteht eine Stromquelle, die durch ihre Leerlaufspannung V_{oc}, ihren Kurzschlussstrom I_{sc} und ihren Punkt größtmöglicher Leistung (Maximum Power Point, MPP) – charakterisiert durch die Größen V_{mpp} und I_{mpp} – gekennzeichnet ist. Die

Ausgangsspannung V des Moduls kann dabei als Funktion des Ausgangsstromes I betrachtet werden [Agilent SAS HB, p. 95] (vgl. auch Abschnitt 3.2.2). Der Kurzschlussstrom I_{sc} ist abhängig von der Intensität der Sonneneinstrahlung: Je intensiver die Sonneneinstrahlung, desto größer der Kurzschlussstrom und somit die MPP-Leistung. Eine beispielhafte Strom-Spannungs- und Leistungs-Spannungs-Kennlinie ist in **Abbildung 2.2** gegeben.

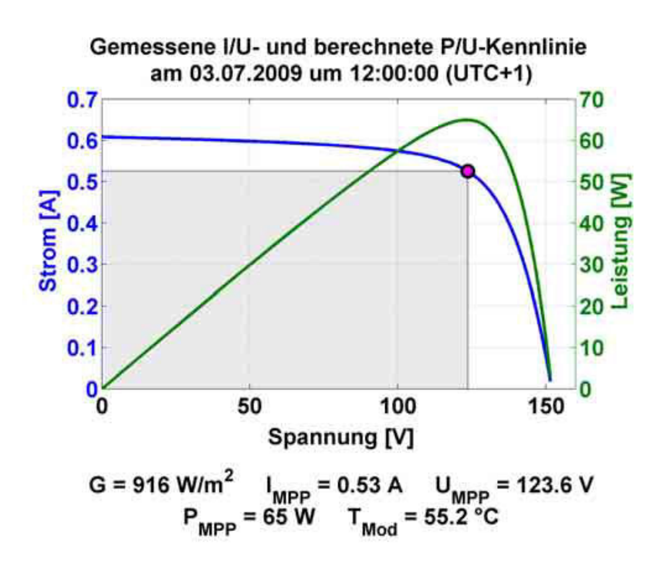


Abbildung 2.2: *Beispielhafte I-V- und P-V-Kennlinien. Aus [Glotzbach 2009, S. 20]*

Typische Solarmodul-Leerlaufspannungen liegen im Bereich von 24 V bis 36 V, in seltenen Fällen auch bei 70 V und darüber. Um durch einen nachgestellten Wechselrichter in das 230 V-Netz einspeisen zu können, ist am Eingang des Wechselrichters jedoch eine Spannung von mindestens $\sqrt{2} \cdot 230\ \text{V} \approx 330\ \text{V}$ (Spitzenwert der Spannung bei einphasiger Einspeisung) wünschenswert. Um diesen Wert zu erreichen, müssen mehrere Module in Serie geschaltet werden.

Da Solaranlagen selten auf freiem Feld aufgebaut werden (können), sondern in den meisten Fällen auf Hausdächern installiert werden, muss mit lokalen Verschattungen gerechnet werden. Dies bedeutet, dass einzelne Solarmodule z. B. im Schatten eines Schornsteins liegen können und somit nicht mit der gleichen Intensität wie die übrigen Module der Serienschaltung bestrahlt werden. Wie bereits vorgängig erläutert, nimmt hierdurch die Ausgangsleistung des Solarmoduls ab.

Treiben die unverschatteten Module ihren maximalen Strom, so wird dieser Strom – als Folge der Serienschaltung – auch durch das verschattete Modul fließen. Je nach Verschattungsgrad ist dieses Modul jedoch nicht mehr in der Lage, einen Strom dieser Stärke zu treiben. Als Folge wird die Spannung gemäß einer Kennlinie wie in **Abbildung 2.2** immer mehr zusammenbrechen, bis sie u. U. schließlich negativ wird.[5] Daraus würde ohne Gegenmaßnahme eine thermische Zerstörung des Moduls und im schlimmsten Falle ein Brand resultieren.

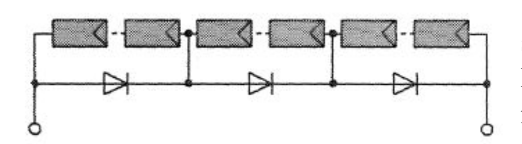


Abbildung 2.3: *Schaltung mit passiven Bypass-Dioden. Aus [Kranzer 2010, p. 3]*

Die einfachste Lösung für dieses Problem ist der Einsatz aktiver oder passiver Bypass-Dioden (**Abbildung 2.3**). Passive Bypass-Dioden sind einfache Dioden, die bei einer negativen Spannung über einem Modul dessen Strom übernehmen und dieses somit faktisch kurzschließen. Dadurch wird das betroffene Modul vollständig aus der Serienschaltung entfernt; die Leistung, die es noch erzeugen könnte, geht verloren.

Da gemäß [Kranzer 2010, p. 3] die passiven Bypass-Dioden jedoch im Bezug auf Kühlung, Leckströme und Defekte durch Überspannung Schwierigkeiten bereiten, werden vermehrt auch aktive Bypass-Dioden in Form von Halbleiter-Schaltern eingesetzt. Diese aktiven Bypass-Dioden weisen bezogen auf die Schwierigkeiten der passiven Bypass-Dioden bessere Eigenschaften auf, führen jedoch ebenfalls zum Verlust der gesamten Modul-Leistung.

Um auch die Leistung der verschatteten Module nutzen zu können, wurden verschiedene Konzepte entwickelt, die in Abschnitt 2.2 kurz vorgestellt werden.

[5] Dies entspricht einer Verlagerung des Arbeitspunktes in den zweiten Quadranten.

2.2 Verschaltung von Solarmodulen

2.2.1 Serienschaltung mit nachgestelltem Wechselrichter

Bleibt man dem Konzept der Serienschaltung von Solarmodulen und anschließender Einspeisung ins Netz mittels Wechselrichter treu, will aber dennoch auch die Leistung von teilweise verschatteten Modulen möglichst vollständig nutzen können, so bietet es sich an, jedes Solarmodul mit einem Gleichspannungswandler auszurüsten. Dieser Gleichspannungswandler wird so angeschlossen, dass der Ausgang eines Solarmoduls direkt mit dem Eingang eines Gleichspannungswandlers verbunden wird; anschließend werden alle Ausgänge der Gleichspannungswandler in Serie geschaltet. Es resultiert eine Schaltung wie in **Abbildung 2.5(a)** (S. 9) dargestellt.

Die Gleichspannungswandler ermöglichen – je nach Ausführung – eine Verschiebung des Arbeitspunktes auf einer Kurve konstanter Leistung.[6] Demzufolge ist es möglich, den Strom der gesamten Serienschaltung von Solarmodulen so einzustellen, dass alle Solarmodule ihre maximale Leistung (MPP-Leistung) einspeisen können. Für die drei verschiedenen Ausführungen von Gleichspannungswandlern gibt es drei entsprechende Anpassungsszenarien:

Werden *Tiefsetzsteller* (engl. *Buck-Converter*) verwendet, so gilt für die Spannung $V_{\text{c,out}}$ am Konverterausgang im Verhältnis zur Ausgangsspannung $V_{\text{PV,out}}$ des Solarmoduls[7] $V_{\text{c,out}} \leq V_{\text{PV,out}}$ und somit $I_{\text{c,out}} \geq I_{\text{PV,out}}$.[8] Um es allen Modulen zu ermöglichen, ihre volle Leistung einzuspeisen, müssen bei diesem Konzept jene Konverter ihre Ausgangsspannung absenken, die an verschatteten Solarmodulen angeschlossen sind. Auf der Kurve konstanter Leistung (rot gestrichelt in **Abbildung 2.4**) können also jene Strom / Spannungs-Werte am Konverterausgang erreicht werden, die *links* von der vertikalen gestrichelten Linie ($V = V_{\text{mpp}} = \text{const.}$) liegen.

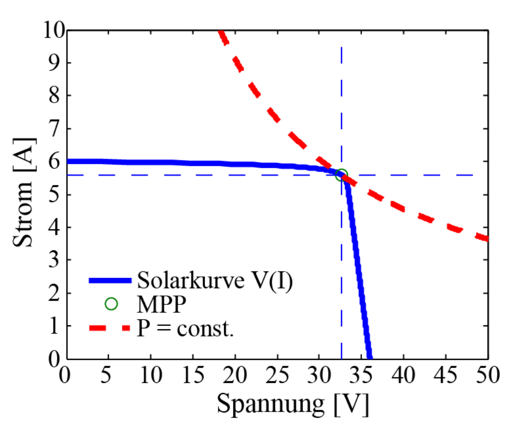


Abbildung 2.4: *Beispielhafte Solarkurve mit Kurve konstanter Leistung.* $V_{oc} = 36$ V, $I_{sc} = 6$ A, $V_{mpp} \approx 33$ V, $I_{mpp} \approx 5.5$ A

Hochsetzsteller (engl. *Boost-Converter*) können am Ausgang nur gleiche oder höhere Spannungen erzeugen als am Eingang anliegen. Demzufolge gilt (mit den gleichen Annahmen wie beim Tiefsetzsteller) $V_{\text{c,out}} \geq V_{\text{PV,out}}$ und $I_{\text{c,out}} \leq I_{\text{PV,out}}$. Somit sind auf der Kurve konstanter Leistung in **Abbildung 2.4** all jene Strom- / Spannungswerte am Konverterausgang einstellbar, die *rechts* der vertikalen gestrichelten Linie liegen. In diesem Falle müssen die Konverter jener Solarmodule, die *nicht* verschattet sind, eine höhere Ausgangsspannung erzeugen, um den Ausgangsstrom auf das Niveau des am stärksten verschatteten Solarmoduls zu senken.

Kommen *Tief-Hochsetzsteller* (engl. *Buck-Boost-Converter*) zum Einsatz, so können (im Rahmen der Konverterspezifikationen) am Konverterausgang beliebige Strom-Spannungspaare, deren Produkt auf der Kurve konstanter Leistung liegt, erreicht werden. Dies bietet die größtmögliche Flexibilität und die (theoretische) Möglichkeit, die Konverter so zu betreiben, dass der Wirkungsgrad der gesamten Anlage optimiert wird. Ein weiterer Vorteil liegt darin, dass die Eingangsspannung des Wechselrichters (auch *Bus-Spannung* genannt) nahezu beliebig frei gewählt und eingestellt werden kann.

[6] Für die elektrische Leistung gilt bekanntlich $P = V \cdot I$, die Vergrößerung von I und Verkleinerung von V respektive umgekehrt kann also so durchgeführt werden, dass die Leistung konstant bleibt.

[7] Es wird angenommen, dass die Verluste zwischen Ausgang des Solarmoduls und Eingang des Konverters vernachlässigbar sind; also $V_{\text{c,in}} = V_{\text{PV,out}}$.

[8] Der Fall der Gleichheit von Eingangs- und Ausgangsspannung ist nur theoretisch und bei einer Effizienz von 100% zu erreichen.

Auf dem Markt sind derzeit alle Konvertertopologien vertreten, wobei einige Hersteller einen zusätzlichen Durchschalt-Modus anbieten: Wird am Konverterausgang in etwa die gleiche Spannung benötigt wie sie am Konvertereingang anliegt, so werden Konverterein- und -ausgang direkt miteinander verbunden. Durch diese zusätzliche Maßnahme können die Schaltverluste in diesem Modus eliminiert werden, einzig die Leitungsverluste der Konverter (in erster Linie der Schalter) halten in diesem Falle die Effizienz unter 100%.

In **Tabelle 2.1** ist eine Übersicht über die derzeit auf dem Markt verfügbaren Gleichspannungswandler für Serienschaltung der Ausgänge gegeben.

Typ	*Hersteller und Modellbezeichnung*	*Spezifikationen*
Tiefsetzsteller	Azuray AP250	$V_{in,min} = 20$ V, $V_{in,max} = 80$ V $I_{in,typ} = 8$ A, $I_{in,max} = 10$ A $P_{in,max} = 250$ W $V_{out,max} = V_{PV,out}$, $I_{out,max} = 14$ A
Tiefsetzsteller	Azuray AP260	$V_{in,min} = 8$ V, $V_{in,max} = 70$ V $I_{in,typ} = 8$ A, $I_{in,max} = 10$ A $P_{in,max} = 300$ W $V_{out,max} = V_{PV,out}$, $I_{out,max} = 15$ A
Tiefsetzsteller / Durchschalt-Modus	Azuray AP300	$V_{in,min} = 8$ V, $V_{in,max} = 80$ V $I_{in,typ} = 8$ A, $I_{in,max} = 10$ A $P_{in,max} = 300$ W $V_{out,max} = V_{PV,out}$, $I_{out,max} = 15$ A
Tiefsetzsteller	Tigo MM-ES50	$V_{in,max} = 52$ V, $V_{in,mpp} \in [16\text{ V}, 48\text{ V}]$ $I_{in,max} = 10$ A, $I_{in,mpp} = 9{,}5$ A $P_{in,max} = 350$ W $V_{out,max} = V_{PV,out}$, $I_{out,max} = 9{,}5$ A
Tiefsetzsteller	Tigo MM-ES75	$V_{in,max} = 75$ V, $V_{in,mpp} \in [30\text{ V}, 65\text{ V}]$ $I_{in,max} = 7{,}5$ A, $I_{in,mpp} = 6{,}5$ A $P_{in,max} = 350$ W $V_{out,max} = V_{PV,out}$, $I_{out,max} = 6{,}5$ A
Tiefsetzsteller	Tigo MM-ES110	$V_{in,max} = 110$ V, $V_{in,mpp} \in [30\text{ V}, 89\text{ V}]$ $I_{in,max} = 5$ A, $I_{in,mpp} = 4{,}7$ A $P_{in,max} = 350$ W $V_{out,max} = V_{PV,out}$, $I_{out,max} = 4{,}7$ A
Tiefsetzsteller	Tigo MM-ES170	$V_{in,max} = 170$ V, $V_{in,mpp} \in [30\text{ V}, 140\text{ V}]$ $I_{in,max} = 3$ A, $I_{in,mpp} = 2{,}6$ A $P_{in,max} = 350$ W $V_{out,max} = V_{PV,out}$, $I_{out,max} = 2{,}6$ A
Hochsetzsteller	ST Microelectronics SPV 1020[9]	$V_{in,max} = 40$ V, $V_{in,min} = 6{,}5$ V $I_{in,max} = 9$ A $P_{max} = 300$ W $V_{out,max} = 36$ V

[9]Die Angaben beziehen sich auf das Evaluations-Board STEVAL-ISV009V1, welches auf dem IC SPV1020 und der Bypass-Diode SPV1001N30 basiert.

Tief-Hochsetzsteller	SolarMagic[10] SM1230-xAx	$V_{\text{in,max}} = 50$ V, $V_{\text{in,mpp}} \leq 40$ V $I_{\text{PV,sc}} \in [9{,}2\ \text{A}, 10{,}4\ \text{A}]$, $I_{\text{in,mpp}} \leq 8{,}5$ A $V_{\text{out,max}} = 43$ V, $I_{\text{out,max}} = 8{,}5$ A
Tief-Hochsetzsteller	SolarMagic SM1230-xBx	$V_{\text{in,max}} = 100$ V, $V_{\text{in,mpp}} \leq 80$ V $I_{\text{PV,sc}} \in [5{,}9\ \text{A}, 6{,}4\ \text{A}]$, $I_{\text{in,mpp}} \leq 8{,}5$ A $V_{\text{out,max}} = 80$ V, $I_{\text{out,max}} = 5{,}5$ A
Tief-Hochsetzsteller / Durchschalt-Modus	SolarMagic SM3320-xAx	$V_{\text{in,max}} = 50$ V, $V_{\text{in,mpp}} \leq 40$ V $I_{\text{in,max}} = 11$ A, $I_{\text{in,mpp}} \leq 11$ A $V_{\text{out,max}} = 43$ V, $I_{\text{out,max}} = 12{,}5$ A
Tief-Hochsetzsteller / Durchschalt-Modus	SolarMagic SM3320-xBx	$V_{\text{in,max}} = 80$ V, $V_{\text{in,mpp}} \leq 42$ V $I_{\text{in,max}} = 9$ A, $I_{\text{in,mpp}} \leq 9$ A $V_{\text{out,max}} = 46$ V, $I_{\text{out,max}} = 10$ A
Tief-Hochsetzsteller / Durchschalt-Modus	SolarMagic SM3320-xCx	$V_{\text{in,max}} = 80$ V, $V_{\text{in,mpp}} \leq 50$ V $I_{\text{in,max}} = 9$ A, $I_{\text{in,mpp}} \leq 9$ A $V_{\text{out,max}} = 53$ V, $I_{\text{out,max}} = 10$ A
Tief-Hochsetzsteller / Durchschalt-Modus	SolarMagic SM3320-xDx	$V_{\text{in,max}} = 100$ V, $V_{\text{in,mpp}} \leq 70$ V $I_{\text{in,max}} = 9$ A, $I_{\text{in,mpp}} \leq 9$ A $V_{\text{out,max}} = 74$ V, $I_{\text{out,max}} = 10$ A
Tief-Hochsetzsteller	SolarEdge PB250-AOB	$V_{\text{in,max}} = 60$ V, $V_{\text{in,mpp}} \in [5\ \text{V}, 60\ \text{V}]$ $I_{\text{in,max}} = 10$ A $P_{\text{in,max}} = 250$ W $V_{\text{out}} \in [5\ \text{V}, 60\ \text{V}]$, $I_{\text{out,max}} = 15$ A
Tief-Hochsetzsteller	SolarEdge PB350-AOB	$V_{\text{in,max}} = 60$ V, $V_{\text{in,mpp}} \in [5\ \text{V}, 60\ \text{V}]$ $I_{\text{in,max}} = 10$ A $P_{\text{in,max}} = 350$ W $V_{\text{out}} \in [5\ \text{V}, 60\ \text{V}]$, $I_{\text{out,max}} = 15$ A

Tabelle 2.1: *Kommerzielle Gleichspannungswandler für Serienschaltungen auf dem Markt. Die zugehörigen Datenblätter sind im Literaturverzeichnis (S. 46) speziell erwähnt.*

2.2.2 Direkte Netzeinspeisung

Mittels potentialgetrennter Wechselrichter ist es möglich, vom Ausgang *eines* Solarmoduls direkt ins Netz einzuspeisen. Dabei gibt es verschiedene Konvertertypen, die zur Anwendung gelangen. Die Potentialtrennung ist jedoch in jedem Falle notwendig, da ohne Transformator eine derart große Spannungsdifferenz zwischen Solarmodulausgang und Konverterausgang nicht realisierbar ist. In den meisten Fällen kommen Hochfrequenztransformatoren zum Einsatz, die gegenüber den Niederfrequenztransformatoren eine deutlich geringere Baugröße und Masse aufweisen.

Nachteil dieser Topologie ist die – durch die Transformatoren bedingte – relativ geringe Effizienz von höchstens 95% [Kranzer 2010, p. 25-28]. Dennoch werden einige sogenannte Mikro-Inverter kommerziell produziert, in [Kranzer 2010] werden *Dorfmüller*, *Enphase*, *Enecsys* und *SolarBridge* als Hersteller genannt.

Dieser Typ von Konvertern ist für die vorliegende Arbeit nicht von Bedeutung, da das Forschungsinteresse *modulintegrierten* Konvertern gilt. Mit Massen von bis zu 2 kg sind diese Konverter für einen direkten

[10] Teil von National Semiconductor

Einbau am Modul jedoch kaum geeignet; außerdem steht auf Grund ihrer relativ geringen Effizienz zu vermuten, dass sich dieser Konvertertyp als Massenlösung kaum durchsetzen wird.

2.2.3 Parallelschaltung

Ein weiterer auf dem Markt erhältlicher Lösungsansatz verändert die Verschaltung der Konverterausgänge: Anstatt die Ausgänge der modulintegrierten Gleichspannungswandler wie bisher in Serie zu schalten, werden die Ausgänge der modulintegrierten Gleichspannungswandler parallel geschaltet. Die Konverter – z. B. *Sperrwandler* (engl. *Flyback-Converter*) – werden dabei so betrieben, dass am Ausgang die gewünschte Wechselrichter-Eingangsspannung anliegt [Kranzer 2010, p. 21 f.]. Der fließende Strom stellt sich entsprechend der verfügbaren Leistung ein.

Die auf dem Markt erhältlichen Gleichspannungswandler für Parallelschaltung sind in **Tabelle 2.2** kurz vorgestellt; eine beispielhafte Schaltung ist in **Abbildung 2.5(b)** gegeben.

Typ	*Hersteller und Modellbezeichnung*	*Spezifikationen*
Sperrwandler	eIQenergy vBOOST250	$V_{\text{in,min}} = 20$ V, $V_{\text{in,max}} = 50$ V $I_{\text{in,max}} = 10$ A $P_{\text{in,max}} = 250$ W $V_{\text{out}} \in [250\text{ V}, 350\text{ V}]$, $I_{\text{out,max}} = 1{,}25$ A
Sperrwandler	eIQenergy vBOOST350	$V_{\text{in,min}} = 30$ V, $V_{\text{in,max}} = 100$ V $I_{\text{in,max}} = 10$ A $P_{\text{in,max}} = 350$ W $V_{\text{out}} \in [250\text{ V}, 350\text{ V}]$, $I_{\text{out,max}} = 1{,}25$ A
Sperrwandler	Tigo MM-EP35	$V_{\text{in,max}} = 55$ V, $V_{\text{in,mpp}} \in [28\text{ V}, 42\text{ V}]$ $I_{\text{in,max}} = 6{,}5$ A, $I_{\text{in,mpp}} = 5{,}7$ A $P_{\text{in,max}} = 200$ W $V_{\text{out,nom}} = 375$ V, $I_{\text{out,max}} = 0{,}55$ A
Sperrwandler	Tigo MM-EP45	$V_{\text{in,max}} = 54$ V, $V_{\text{in,mpp}} \in [39\text{ V}, 54\text{ V}]$ $I_{\text{in,max}} = 5$ A, $I_{\text{in,mpp}} = 4{,}4$ A $P_{\text{in,max}} = 200$ W $V_{\text{out,nom}} = 375$ V, $I_{\text{out,max}} = 0{,}55$ A
Sperrwandler	Tigo MM-EP35	$V_{\text{in,max}} = 60$ V, $V_{\text{in,mpp}} \in [53\text{ V}, 60\text{ V}]$ $I_{\text{in,max}} = 4$ A, $I_{\text{in,mpp}} = 3{,}3$ A $P_{\text{in,max}} = 200$ W $V_{\text{out,nom}} = 375$ V, $I_{\text{out,max}} = 0{,}55$ A

Tabelle 2.2: *Kommerzielle Gleichspannungswandler für Parallelschaltung auf dem Markt. Die zugehörigen Datenblätter sind im Literaturverzeichnis (S. 46) speziell erwähnt.*

2.3 MPP-Tracking

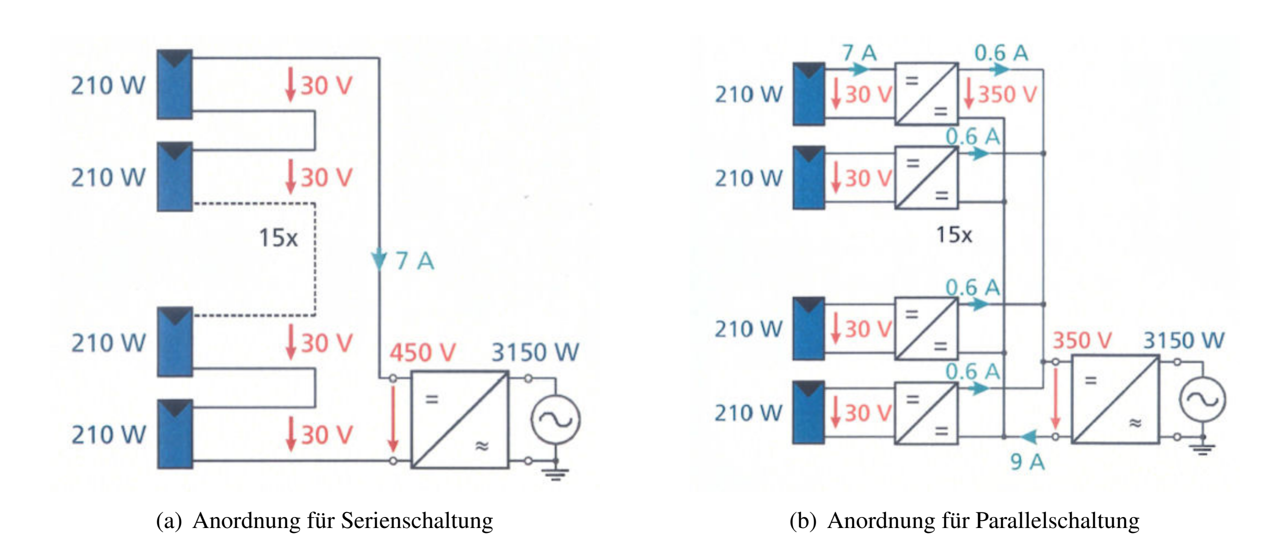
In **Abbildung 2.2** ist ersichtlich, dass die Leistung eines Solarmoduls von der Ausgangsspannung abhängig ist. Jenes Wertepaar von Strom und Spannung, an denen das Produkt $P = I \cdot V$ maximal wird, bezeichnet man als Punkt größtmöglicher Leistung, englisch *Maximum Power Point* oder kurz *MPP*.

Um eine maximale Ausnutzung des Solarmoduls zu erreichen, sollte also die Spannung am Solarmodulausgang (= Konvertereingang) möglichst dem MPP-Wert V_{mpp} entsprechen. Zu diesem Zwecke verfügen die Konverter für den Einsatz an Solarmodulen über sogenanntes MPP-Tracking: Sie „suchen“

nach dem MPP. Dies geschieht durch eine schrittweise Änderung der Spannung am Konvertereingang: Ist die Eingangsleistung des neuen Arbeitspunktes größer als jene am alten Arbeitspunkt, so wird ein weiterer Spannungsschritt in die gleiche Richtung durchgeführt. Ist die neue Eingangsleistung kleiner als die alte, so wird wieder ein Spannungsschritt in die andere Richtung durchgeführt.

Durch dieses „Suchen“ nach dem MPP geht Leistung verloren, da der Konverter während eines Großteils der Zeit außerhalb des MPPs arbeitet. Die Qualität des MPP-Tracking-Algorithmus’ hat also auch einen Einfluss auf die Gesamteffizienz des Konverters.

In Kapitel 5 wird im Rahmen der Ergebnisse einiger Testmessungen nochmals auf das MPP-Tracking eingegangen.



(a) Anordnung für Serienschaltung

(b) Anordnung für Parallelschaltung

Abbildung 2.5: *Beispiele für Verschaltungen von Solarmodulen mit Gleichspannungswandlern. Aus [Kranzer 2010, p. 5] (a) respektive [Kranzer 2010, p. 24] (b)*

3 Aufbau des Teststandes

3.1 Versuchsanordnung

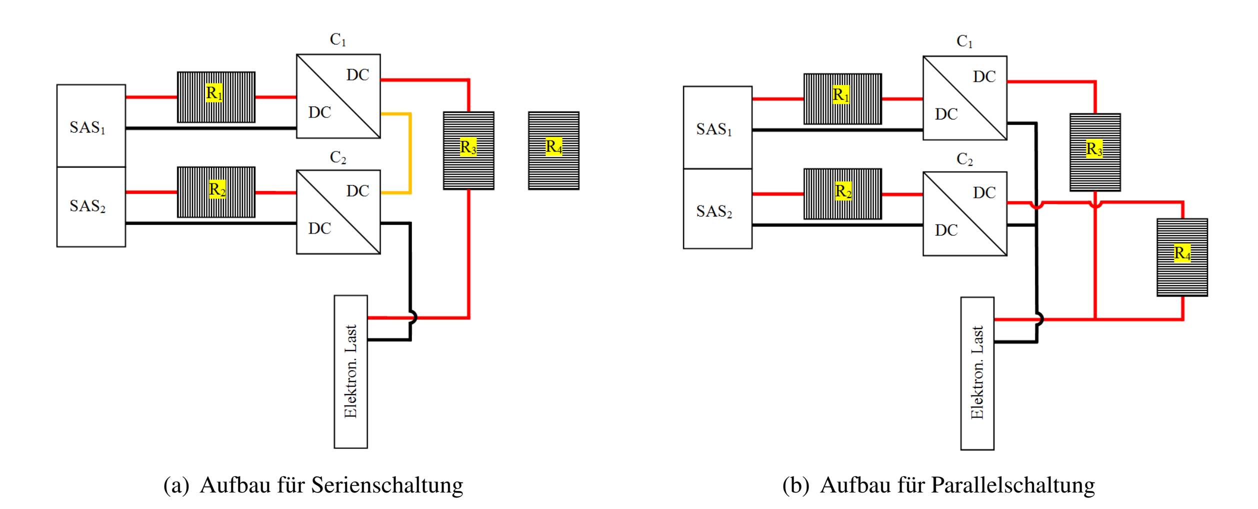
Ziel der Versuchsanordnung ist es, solarmodulintegrierte Konverter auf ihre Effizienz hin untersuchen zu können.

Da die Konverter bereits heute Effizienzen von 95% und darüber erreichen [Kranzer 2010], sind hochgenaue Messgeräte zu verwenden sowie im Aufbau der Schaltung parasitäre Induktivitäten, Widerstände etc. möglichst gering zu halten. (Auf die Messgenauigkeit wird in Abschnitt 3.5 genauer eingegangen.)

Weiterhin wurden in Abschnitt 2.2 zwei verschiedene Verschaltungsarten von Solarmodulen vorgestellt. Der Teststand soll also so aufgebaut werden, dass mit möglichst wenig Umsteck-Aufwand Konverter für beide Arten der Verschaltung getestet werden können.

Wie der Solar Array Simulator, der Solarmodule simuliert, mit den Konvertern und diese mit der Last verbunden werden müssen, ergibt sich aus den in Abschnitt 2.2 erörterten Schaltungsvarianten. Für eine genauere Betrachtung der verschiedenen Konvertertypen, ihres Verhaltens und ihrer Verschaltung sei auf [Kranzer 2010] verwiesen.

Der resultierende schematische Aufbau des Teststandes, betrachtet aus der Vogelperspektive, ist in **Abbildung 3.1** für beide Schaltungsarten dargestellt. **Abbildung 3.2** zeigt den Schaltplan des Teststandes mit allen Mess-Abgriffen, ebenfalls für beide Schaltungsarten.



(a) Aufbau für Serienschaltung

(b) Aufbau für Parallelschaltung

Abbildung 3.1: *Schematischer Aufbau des Teststandes für Serien- und Parallelschaltung der Solarmodule; von oben betrachtet. Legende:*

SAS_1, SAS_2: Solar Array Simulator-Ausgänge, siehe Abschnitt 3.2.2

R_1 bis R_4: Hochpräzisionswiderstände als Shunts, siehe Abschnitt 3.2.4

C_1 und C_2: Gleichspannungswandler; Prüflinge, siehe Abschnitt 3.3

Elektron. Last: Elektronische Last als Verbraucher, siehe Abschnitt 3.2.6

Rote / orange / schwarze Verbindungen: Kabel auf hohem / mittlerem / niedrigem Potential

Die Messverbindungen sowie die Messgeräte sind nicht eingezeichnet.

3.2 Messgeräte

3.2.1 Überblick

Um den Teststand gemäß **Abbildung 3.2** aufzubauen, werden insgesamt 14 Geräte benötigt: Ein Solar Array Simulator, sieben Tischmultimeter zur Spannungsmessung, vier Hochpräzisionswiderstände als Shunts, ein Funktionsgenerator als Trigger sowie eine elektronische Last. **Tabelle 3.1** gibt einen Überblick über die während des ersten Aufbaus des Teststandes verwendeten Geräte und ihre Seriennummern zwecks eindeutiger Identifizierung.[11]

*Bezeichnung (in **Abb. 3.2**)*	*ID*[12]	*Gerätebezeichnung*	*Seriennummer*[13]
SAS (SAS_1, SAS_2)	9	Agilent E4360A	MY48002120
V_1	4	Agilent 34410A	MY47024632
V_2	3		MY47024592
V_3	2		MY47024612
V_4	7		MY47023453
V_5	6		MY47023452
V_6	5		MY47023020
V_7	1		MY47024627
R_1	-	burster 1282 10 mΩ	7320
R_2	-		7321
R_3	-		7322
R_4	-		5162
Trig	8	Agilent 33220A	MY44012795
Electronic Load	10	Chroma 63202 2,6 kW	AGZ000000219

Tabelle 3.1: *Übersicht über die verwendeten Geräte und ihre Seriennummern*

Die Gleichspannungswandler C_1 und C_2 in **Abbildung 3.2** sind die auszumessenden Prüflinge.

3.2.2 Solar Array Simulator „Agilent E4360A“

Zum Aufbau des Teststandes kommt der Solar Array Simulator *Agilent E4360A* mit SAS-Modulen des Typs *Agilent E4361A-J01* zum Einsatz.

Die zwei SAS-Module werden im Folgenden mit SAS_1 und SAS_2 bezeichnet, ihre Spezifikationen bei einer Netz-Nennspannung von $V_{\text{eff}} = 230\ \text{V}\sim$ sind in Anhang C in Tabelle C.1 gegeben.

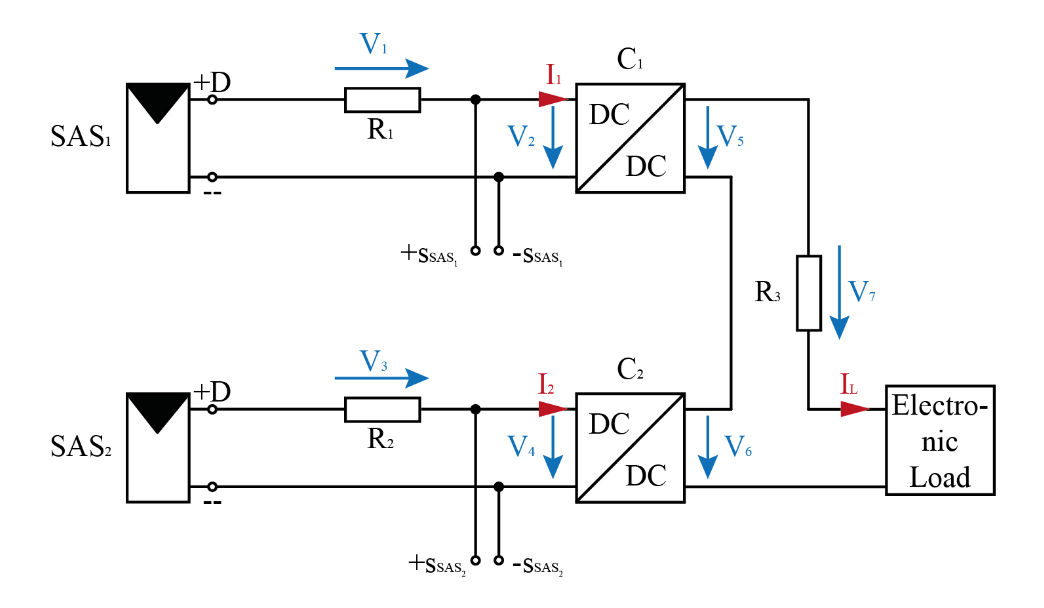
Der SAS unterstützt drei verschiedene Betriebsmodi:

Fixed Mode Im *Fixed Mode*, kurz FIX, verhält sich der SAS wie eine ideale Spannungsquelle: Für jeden beliebigen Strom unterhalb des definierten Grenzwertes steht die definierte Ausgangsspannung zur Verfügung.

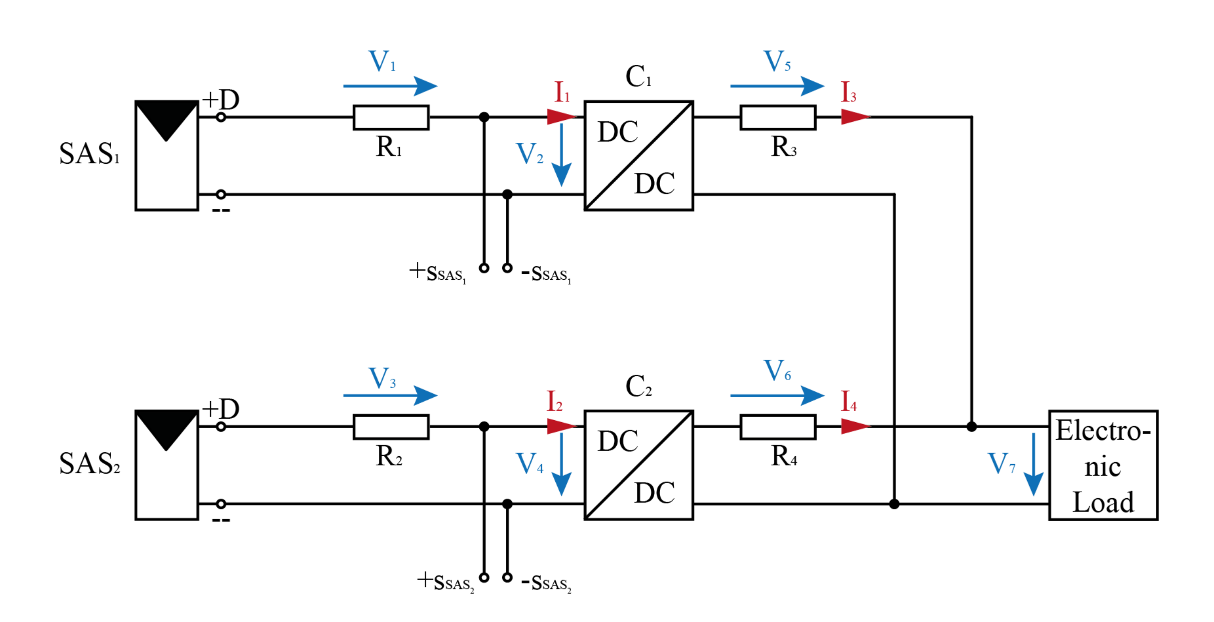
[11] Zum Zeitpunkt der Abgabe dieser Arbeit waren alle Geräte wie hier beschrieben im Einsatz. Allerdings ist es möglich, dass im späteren Einsatz das eine oder andere Gerät ausgetauscht wird. In diesem Falle sind einige Anpassungen nötig, vgl. hierzu Anhang B.

[12] Unter der ID wird die programminterne Identifikationsnummer verstanden.

[13] Bei den Widerständen R_1 bis R_4 ist an Stelle der Seriennummer die Nummer des Kalibrierungszertifikats angegeben.



(a) Schaltplan für Serienschaltung der Gleichspannungswandler



(b) Schaltplan für Parallelschaltung der Gleichspannungswandler

Abbildung 3.2: *Schaltpläne des Teststandes für Serien- und Parallelschaltung der Gleichspannungswandler*

SAS Mode Im *SAS-Modus* berechnet der SAS anhand der Vorgabe der Parameter V_{oc}, I_{sc}, V_{mpp} und I_{mpp} eine Solarkurve und simuliert das Verhalten eines Solarmoduls. Auf das verwendete Modell wird später in diesem Abschnitt genauer eingegangen.

Table Mode Im *Table Mode*, kurz TABL, simuliert der SAS anhand einer gespeicherten Tabelle das Verhalten eines Solarmoduls.

Für den Teststand wird hauptsächlich der SAS-Modus benötigt, um möglichst flexibel das Verhalten des Solarmoduls einstellen zu können. Zu Testzwecken wird ebenfalls der Modus FIX unterstützt.

SAS-Mode-Modell Laut [Agilent SAS HB, p. 58] verwendet der SAS im SAS-Modus zur Simulation des Solarmoduls ein exponentielles Modell, das durch die Parameter V_{oc}, I_{sc}, V_{mpp} und I_{mpp} definiert wird. In [Agilent SAS HB, p. 60] wird explizit darauf hingewiesen, dass im Falle $I_{mpp} \ll I_{sc}$ eine Abweichung des durch die Parameter definierten MPP_{def} vom real simulierten MPP_{real} auftreten kann.

Verwendet man die Modellgleichungen aus [Agilent SAS HB, p. 95], so ist es möglich, das Modell in der Steuerungssoftware des Teststandes nachzubilden. Auf die Behandlung dieser Abweichung des definierten MPP vom real simulierten MPP wird in Abschnitt 4.4 eingegangen; an dieser Stelle sollen zuerst die dem Modell zu Grunde liegenden Gleichungen angegeben werden:

$$V(I) = \frac{\frac{V_{oc} \cdot \ln\left(2 - \left(\frac{I}{I_{sc}}\right)^N\right)}{\ln(2)} - R_s \cdot (I - I_{sc})}{1 + \frac{R_s \cdot I_{sc}}{V_{oc}}}, \tag{3.1}$$

wobei

$$R_s = \frac{V_{oc} - V_{mpp}}{I_{mpp}}, \tag{3.2a}$$

$$N = \frac{\ln(2 - 2^a)}{\ln\left(\frac{I_{mpp}}{I_{sc}}\right)} \quad \text{und} \tag{3.2b}$$

$$a = \frac{V_{mpp} \cdot \left(1 + \left(\frac{R_s \cdot I_{sc}}{V_{oc}}\right)\right) + R_s \left(I_{mpp} - I_{sc}\right)}{V_{oc}}. \tag{3.2c}$$

Abbildung 3.4(a) zeigt ein Beispiel für die Abweichung des real simulierten MPPs vom definierten MPP: Sowohl Strom, Spannung als auch Leistung entsprechen *nicht* den definierten Werten ($P_{real} \approx 1{,}05 \cdot P_{def} \not\approx P_{real}$); **Abbildung 3.4(b)** zeigt, dass das Modell für $I_{mpp} \lesssim I_{sc}$ den MPP jedoch in guter Näherung trifft ($P_{real} \approx 1{,}0003 \cdot P_{def} \approx P_{real}$).

Anschlüsse des SAS-Ausgangs Jedes SAS-Modul verfügt am Ausgang über fünf Klemmen mit den Bezeichnungen *+D*, *+s*, +, - - und *-s*. **Abbildung 3.3** zeigt eine schematische Darstellung des Ausgangs eines SAS-Moduls.

Bei den Anschlüssen + und - - handelt es sich dabei um die üblichen Ausgänge auf hohem respektive tiefem Potential. Der Anschluss *+D* ist intern über eine Diode mit dem Anschluss + verbunden. Verwendet man diesen Anschluss anstelle des Anschlusses +, so werden (u. U. gefährliche) Rückströme unterbunden.

Bei den Anschlüssen *+s* und *-s* handelt es sich um Mess-*Eingänge*, die bei längeren Distanzen oder Verlusten zwischen SAS-Ausgang und Verbrauchereingang dazu verwendet werden können, die Verbrauchereingänge zurückzuführen. Der SAS wird dann die Ausgänge so regeln, dass am Verbrauchereingang stets die programmierte Kurve zur Verfügung steht. Die am Ausgang anliegenden Strom- / Spannungswerte können ausgelesen werden.

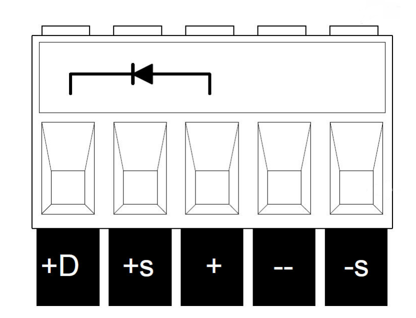


Abbildung 3.3: *Ausgang eines SAS-Moduls.*
Aus [Agilent SAS HB, p. 21]

Fernzugriff Der SAS verfügt über eine USB-Schnittstelle, über die das Gerät mittels SCPI-Befehlen ferngesteuert werden kann. Die Hersteller-ID von Agilent lautet `0x0957`, die Geräte-ID lautet `0x1107`. Die Funktionsweise des Fernzugriffs wird in Abschnitt 4.3 erläutert.

3.2.3 Tischmultimeter „Agilent 34410A"

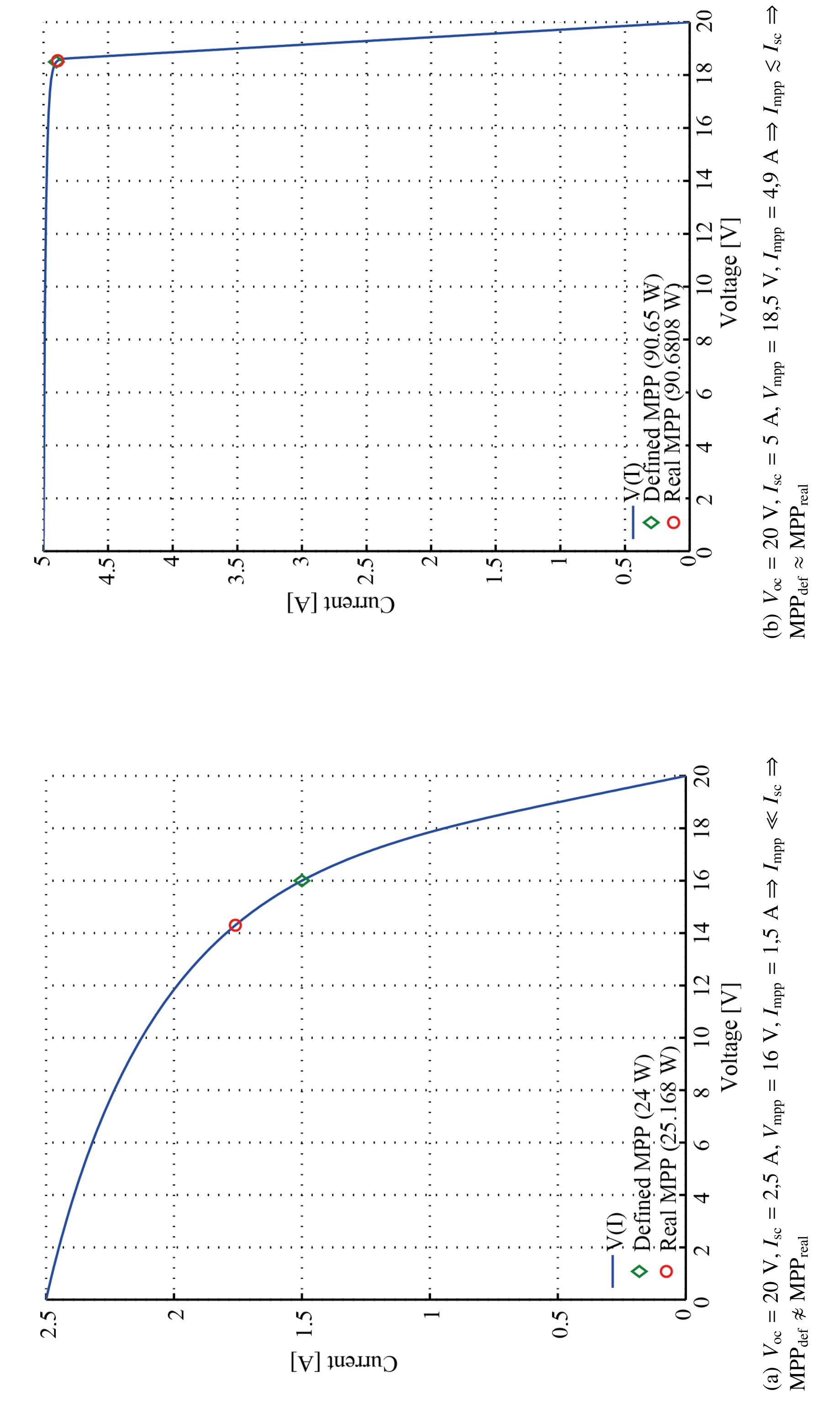
Bei diesem Tischmultimeter handelt es sich um ein $6^1/_2$ digit-Präzisionsmessgerät. Es unterstützt neben Gleichstrom- und Gleichspannungsmessungen viele weitere Messarten, die für diesen Teststand jedoch nicht von Bedeutung sind. Die verfügbaren Spannungsmessbereiche sind 100 mV, 1 V, 10 V, 100 V und 1000 V; für Gleichstrommessungen stehen die Messbereiche 100 μA, 1 mA, 10 mA, 100 mA, 1 A und 3 A zur Verfügung. Auf Grund der Tatsache, dass der größte Messbereich für Gleichstrom 3 A beträgt, die Ausgangsströme von Solarmodulen allerdings bis 10 A und sogar darüber gehen können, ist der Einsatz von Shunts zur Strommessung nötig (siehe **Tabellen 2.1**, **2.2**).[14] Weiterhin ist auf Grund der Spezifikationen in [Agilent TMM HB, S. 143] eine um Faktoren schlechtere Messgenauigkeit zu erwarten.

Für Gleichspannungsmessungen stellt das Multimeter zwei verschiedene Arten der Integrationszeit-Angabe zur Verfügung: Entweder, es wird direkt die Integrationszeit angegeben, oder es wird eine Integrationszeit relativ zur Netzzykluszeit angegeben. Im zweiteren Falle wird der Faktor, mit dem die Netzzykluszeit multipliziert werden soll, mit NPLC[15] bezeichnet; NPLC $\in$ {0,006, 0,02, 0,06, 0,2, 1, 2, 10, 100}.
Je nach gewählter NPLC verändert sich die Messauflösung für die Messung. Eine Zusammenstellung ist in Anhang C in Tabelle C.3 gegeben; die wichtigsten technischen Daten der Messgeräte sind in Tabelle C.2 aufgeführt.

Um die Messgeräte möglichst synchron starten zu können, steht ein BNC-Eingang für ein externes Triggersignal zur Verfügung. Das Triggersignal muss eine Amplitude von 3 V bis 5 V und eine Dauer von mindestens 1 μs aufweisen [Agilent TMM HB, S. 83]. Es ist möglich, bis zu 50 000 Messwerte pro Triggersignal aufzunehmen.

Fernzugriff Die Multimeter verfügen über eine USB-Schnittstelle, über die die Geräte mittels SCPI-Befehlen ferngesteuert werden können. Die Hersteller-ID von Agilent lautet `0x0957`, die Geräte-ID lautet `0x0607`.

[14] Mit der Spannung V_R über einem Widerstand mit Widerstandswert R ergibt sich der Strom durch den Widerstand zu $I = V_R/R$.
[15] Number of Power Line Cycles = Anzahl Netzzyklen



(a) V_{oc} = 20 V, I_{sc} = 2,5 A, V_{mpp} = 16 V, I_{mpp} = 1,5 A $\Rightarrow I_{mpp} \ll I_{sc} \Rightarrow$ $\mathrm{MPP_{def}} \not\approx \mathrm{MPP_{real}}$

(b) V_{oc} = 20 V, I_{sc} = 5 A, V_{mpp} = 18,5 V, I_{mpp} = 4,9 A $\Rightarrow I_{mpp} \lesssim I_{sc} \Rightarrow$ $\mathrm{MPP_{def}} \approx \mathrm{MPP_{real}}$

Abbildung 3.4: *Modellkurven des SAS gemäß Gleichung 3.1 zur Illustration der Modellgenauigkeit.*

3.2.4 burster Hochpräzisionswiderstand, Modell 1282

In Abschnitt 3.2.3 wurde festgestellt, dass zur Strommessung Shunts benötigt werden. Zu diesem Zwecke werden burster Hochpräzisionswiderstände des Modells 1282 mit einem Nennwiderstand von 10 mΩ verwendet.

Diese Widerstände verfügen über je zwei Anschlüsse (Eingang und Ausgang) zur Stromführung sowie zwei Anschlüsse zur Spannungsmessung. Sie wurden bei einer Umgebungstemperatur von 23°C kalibriert; ihre Temperaturabhängigkeit ist gegeben als

$$R_T = R_{T=20°\mathrm{C}} \cdot \left(1 + \alpha \cdot (T - 20°\mathrm{C}) + \beta \cdot (T - 20°\mathrm{C})^2\right) \tag{3.3}$$

mit $\alpha \in [0, 20 \cdot 10^{-6}]$ (somit $\overline{\alpha} = 10^{-5}$) und $\beta = -0{,}59 \cdot 10^{-6}$ die thermischen Koeffizienten.

Weitere technische Daten der verwendeten Widerstände sind in Anhang C in Tabelle C.5 gegeben.

3.2.5 Funktionsgenerator „Agilent 33220A“

Der Funktionsgenerator dient in diesem Messaufbau als Trigger. Verwendet wird ein Puls-Signal, welches mittels BNC-Kabel an die Eingänge für einen externen Trigger der Messgeräte übertragen wird. Auf diese Weise wird sichergestellt, dass die Multimeter nahezu perfekt synchron mit ihren Messungen beginnen: Da sich elektromagnetische Wellen und somit elektrische Signale mit Lichtgeschwindigkeit ausbreiten und die Kabel eine Länge zwischen ungefähr 0,5 m und 2 m aufweisen ($\Delta l = 1{,}5$ m), ergibt sich mit $\Delta t = \Delta l / c$ eine vernachlässigbare Laufzeitdifferenz von 5 ns.

Fernzugriff Der Funktionsgenerator verfügt über eine USB-Schnittstelle, über die das Gerät mittels SCPI-Befehlen ferngesteuert werden kann. Die Hersteller-ID von Agilent lautet `0x0957`, die Geräte-ID lautet `0x0407`.

3.2.6 Elektronische Last „Chroma 63202 2,6 kW“

Die elektronische Last dient dazu, den eigentlich an den Ausgang einer Solarmodul-Schaltung angeschlossenen Wechselrichter zu simulieren.

Die Last unterstützt verschiedene Operationsmodi, so *Konstantstrom (CC)*, *Konstantwiderstand (CR)*, *Konstantspannung (CV)* und *Konstantleistung (CP)* sowie einige spezielle Derivate dieser Modi. Ein Wechselrichter wird im Allgemeinen eine konstante Eingangsspannung fordern, weshalb der Modus *CV* der für den Teststand bedeutendste ist. Allerdings werden auch andere Modi implementiert, falls das Verhalten der Konverter unter anderen Bedingungen getestet werden soll.

Zu Informationszwecken ist es möglich, die Momentanwerte von Strom, Spannung und Leistung / Widerstand der Last auszulesen. Leider lässt die Genauigkeit dieser Messwerte zu wünschen übrig, was die Verwendung zur Effizienzbestimmung ausschließt.
Die wichtigsten technischen Daten der elektronischen Last sind in Anhang C in Tabelle C.6 aufgeführt.

Fernzugriff Die Last verfügt über eine GPIB-Schnittstelle, die mittels eines GPIB-Adapters[16] von National Instruments auf eine USB-Schnittstelle am PC übersetzt werden kann. Die Last erscheint dann als GPIB-Device, für den Zugriff sind *Board Address* `0` und *Primary Address* `2` erforderlich. Die Funktionsweise des Fernzugriffs wird in Abschnitt 4.3 erläutert.

[16] National Instruments GPIB-USB-HS

3.3 Prüflinge

Als Prüflinge kommen grundsätzlich alle Gleichspannungswandler für Serien- oder Parallelschaltung in Frage. Für diese Arbeit beschränkt sich die Auswahl der Gleichspannungswandler auf die in den **Tabellen 2.1** und **2.2** aufgeführten Konverter.

Zumal viele der Konverter einzeln nur sehr schwer zu beschaffen sind,[17] wurden die Tests während der Entwicklungsphase mit dem Evaluations-Kit SM3320-RF-EV des Konverters *SolarMagic SM3320-1A1* (vgl. **Tabelle 2.1**) durchgeführt. Gegen Ende der Arbeit gelang es, auch noch das Evaluations-Kit STEVAL-ISV009V1, welches auf dem Hochsetzsteller *ST Microelectronics SPV1020* (vgl. **Tabelle 2.1**) basiert, zu beschaffen.

3.4 Berechnung der Ergebnisse

Das Hauptinteresse des in dieser Arbeit entwickelten Teststandes gilt der Effizienz der zu testenden Wechselrichter, sprich der Prüflinge. Insgesamt sind drei verschiedene Effizienzen von Interesse:

- Die Effizienz des Konverters (Konvertereffizienz) η_c vom Konvertereingang zum Konverterausgang,
- die Effizienz des MPP-Trackings (MPP-Effizienz) η_MPP, sprich das Verhältnis von Konvertereingangsleistung zu verfügbarer MPP-Leistung sowie
- die Gesamteffizienz des Systems η_tot von der MPP-Leistung zur Konverterausgangsleistung.

Die Effizienz eines jeden Systems ist grundsätzlich definiert als

$$\eta := \frac{P_\mathrm{out}}{P_\mathrm{in}} < 100\%. \tag{3.4}$$

Da jedoch weder die Effizienz noch die Leistung mit dem in **Abbildung 3.2** gezeigten Aufbau direkt gemessen werden können, müssen die besagten Effizienzen berechnet werden. Dies geschieht für die Konvertereffizienz gemäß den Gleichungen 3.5 für Serienschaltung und 3.6 für Parallelschaltung, für die MPP-Effizienz gemäß den Gleichungen 3.7 und für die Gesamteffizienz gemäß den Gleichungen 3.8 für Serienschaltung und 3.9 für Parallelschaltung. Die Bezeichnungen der Größen orientieren sich dabei an **Abbildung 3.2**:

$$\eta_\mathrm{c1,S} = \frac{P_\mathrm{c1,out}}{P_\mathrm{c1,in}} = \frac{V_5 \cdot I_\mathrm{L}}{V_2 \cdot I_1} = \frac{V_5 \cdot \frac{V_7}{R_3}}{V_2 \cdot \frac{V_1}{R_1}} \tag{3.5a}$$

$$\eta_\mathrm{c2,S} = \frac{P_\mathrm{c2,out}}{P_\mathrm{c2,in}} = \frac{V_6 \cdot I_\mathrm{L}}{V_4 \cdot I_2} = \frac{V_6 \cdot \frac{V_7}{R_3}}{V_4 \cdot \frac{V_3}{R_2}} \tag{3.5b}$$

$$\eta_\mathrm{c1,P} = \frac{P_\mathrm{c1,out}}{P_\mathrm{c1,in}} = \frac{V_7 \cdot I_3}{V_2 \cdot I_1} = \frac{V_7 \cdot \frac{V_5}{R_3}}{V_2 \cdot \frac{V_1}{R_1}} \tag{3.6a}$$

$$\eta_\mathrm{c2,P} = \frac{P_\mathrm{c2,out}}{P_\mathrm{c2,in}} = \frac{V_7 \cdot I_4}{V_4 \cdot I_2} = \frac{V_7 \cdot \frac{V_6}{R_4}}{V_4 \cdot \frac{V_3}{R_2}} \tag{3.6b}$$

[17] Es konnte ein einziger Anbieter in den USA ermittelt werden, der relativ viele der Konverter im Sortiment hat.

$$\eta_{\text{MPP,c1}} = \frac{P_{\text{c1,in}}}{P_{\text{MPP1,real}}} = \frac{V_2 \cdot I_1}{P_{\text{MPP1,real}}} = \frac{V_2 \cdot \frac{V_1}{R_1}}{P_{\text{MPP1,real}}} \tag{3.7a}$$

$$\eta_{\text{MPP,c2}} = \frac{P_{\text{c2,in}}}{P_{\text{MPP2,real}}} = \frac{V_4 \cdot I_2}{P_{\text{MPP2,real}}} = \frac{V_4 \cdot \frac{V_3}{R_2}}{P_{\text{MPP2,real}}} \tag{3.7b}$$

$$\eta_{\text{tot,c1,S}} = \frac{P_{\text{c1,out}}}{P_{\text{MPP1,real}}} = \frac{V_5 \cdot I_{\text{L}}}{P_{\text{MPP1,real}}} = \frac{V_5 \cdot \frac{V_1}{R_1}}{P_{\text{MPP1,real}}} \tag{3.8a}$$

$$\eta_{\text{tot,c2,S}} = \frac{P_{\text{c2,out}}}{P_{\text{MPP2,real}}} = \frac{V_6 \cdot I_{\text{L}}}{P_{\text{MPP2,real}}} = \frac{V_6 \cdot \frac{V_3}{R_2}}{P_{\text{MPP2,real}}} \tag{3.8b}$$

$$\eta_{\text{tot,c1,P}} = \frac{P_{\text{c1,out}}}{P_{\text{MPP1,real}}} = \frac{V_7 \cdot I_3}{P_{\text{MPP1,real}}} = \frac{V_7 \cdot \frac{V_5}{R_3}}{P_{\text{MPP1,real}}} \tag{3.9a}$$

$$\eta_{\text{tot,c2,P}} = \frac{P_{\text{c2,out}}}{P_{\text{MPP2,real}}} = \frac{V_7 \cdot I_4}{P_{\text{MPP2,real}}} = \frac{V_7 \cdot \frac{V_6}{R_4}}{P_{\text{MPP2,real}}} \tag{3.9b}$$

3.5 Messgenauigkeit

3.5.1 Einführung

Wie bereits in Abschnitt 3.1 beschrieben, erreichen heutige Konverter bereits Effizienzen von 95% und darüber. Es ist somit erforderlich, auch kleine Änderungen der Effizienz präzise bestimmen zu können (z. B. von 95,0% auf 95,1%) – was wiederum hohe Anforderungen an die Messgenauigkeit stellt.

In diesem Abschnitt sollen einige Überlegungen zur Genauigkeit der Messungen angestellt werden, die es im Endeffekt ermöglichen, die Genauigkeit der Messungen abzuschätzen. Ziel ist es, die Ungenauigkeiten so genau wie möglich zu berechnen oder – wo dies nicht möglich ist – abzuschätzen.

3.5.2 Zeitpunkt der Messung

In den Gleichungen 3.5 bis 3.9 wird deutlich, dass für die Berechnung jeder Effizienz bis zu vier verschiedene Messwerte nötig sind, mindestens jedoch zwei.

Erfolgen diese Messungen nicht zeitgleich, so ist eine signifikante Ungenauigkeit der Messung zu erwarten, zumal das in Abschnitt 2.3 erläuterte MPP-Tracking regelmäßig die Eingangsspannung des Konverters ändert.[18]

Um die Messungen so zeitgleich wie möglich zu starten, wird – wie bereits in Abschnitt 3.2.5 erläutert – ein Funktionsgenerator als Trigger eingesetzt. Der Unterschied von 5 ns in der Signallaufzeit vom ersten zum letzten Multimeter ist gegenüber dem geringsten Messintervall von (empirisch) etwa 1 ms sowie gegenüber der Frequenz des MPP-Tracking-Algorithmus' von ca. 1 kHz (= 1 Änderung pro Millisekunde) vernachlässigbar.

Somit kann davon ausgegangen werden, dass der Messfehler auf Grund zeitlicher Verschiebungen der Messungen gegeneinander verschwindend gering ist.

[18]Die Testmessungen haben ergeben, dass die Eingangsspannung etwa mit einer Frequenz von 1 kHz geändert wird.

3.5.3 Unsicherheiten der Geräte

Alle Geräte, egal ob Messgeräte oder Widerstände, unterliegen bei der Fertigung gewissen Toleranzen; bei den Messgeräten begrenzt die Anzahl Bits die Genauigkeit der Analog-Digital-Wandlung. Die exakten Werte, die zur Berechnung der Messungenauigkeiten der einzelnen Geräte benötigt werden, sind in Anhang C aufgelistet. Hier soll nun die Messungenauigkeit auf Grund der Geräte ermittelt werden.

Für die **Messunsicherheit eines beliebigen Voltmeters** gilt gemäß [MacLachlan 2011, p. 5]

$$e_{\text{TMM,abs}} = \pm \left(e_{\text{mess}} \cdot V_{\text{mess}} + e_{\text{Bereich}} \cdot V_{\text{Bereich,max}}\right). \tag{3.10}$$

Dabei ist e_{abs} die absolute Unsicherheit; e_{mess} und e_{Bereich} sind relative Unsicherheiten in Prozent, ihre Werte für die für diesen Teststand verwendeten Multimeter sind in Tabelle C.4 aufgelistet.

Um eine Gesamtunsicherheit zu berechnen, bietet es sich an, nur mit relativen Unsicherheiten zu arbeiten: Werden Messgrößen, die mit relativen, normalverteilten Fehlern behaftet sind, miteinander multipliziert oder durcheinander dividiert, dann ergibt sich die Gesamtunsicherheit durch einfache Addition aller relativen Fehler [VUWien 2012, Abschnitt 2.1]. Deshalb wird der absolute Fehler e_{abs} mittels Division von Gleichung 3.10 durch den Messwert V_{mess} auf diesen bezogen:

$$e_{\text{TMM,rel}} = \frac{e_{\text{TMM,abs}}}{V_{\text{mess}}} = \pm \left(e_{\text{mess}} + \frac{e_{\text{Bereich}} \cdot V_{\text{Bereich,max}}}{V_{\text{mess}}}\right). \tag{3.11}$$

Die **Abweichung eines Widerstandswertes** von seinem Nennwert wird in erster Linie durch zwei Faktoren verursacht: Durch die Verlustleistung, die zur Erwärmung und damit zur Änderung des Widerstandswertes gemäß Gleichung 3.3 führt, sowie durch die Fertigungsungenauigkeit $e_{\text{Fertigung}}$.

Die thermischen Verluste im Widerstand setzen sich zusammen aus der durch den Stromfluss resultierenden Verlustleistung, dem thermischen Widerstand und den Temperaturkoeffizienten. Da für den Temperaturkoeffizienten 1. Ordnung α gemäß Tabelle C.5 in Bezug auf den Temperaturkoeffizienten 2. Ordnung β gilt $\alpha \gg \beta$, kann β für diese Betrachtung vernachlässigt werden. Verwendet man den Mittelwert $\overline{\alpha}$ zur Berechnung der Unsicherheit, so ergibt sich

$$e_{\text{R,rel}} = \underbrace{I_{\text{R}}^2 \cdot R_{\text{nom}}}_{P_{\text{V}}} \cdot \underbrace{R_{\text{th}} \cdot \overline{\alpha}}_{\text{Therm. Verluste}} + e_{\text{Fertigung}}. \tag{3.12}$$

Unter Verwendung der obigen Aussage, dass relative Fehler bei Multiplikation oder Division von fehlerbehafteten Messwerten addiert werden, ist es nun möglich, für jeden Messwert und jede berechnete Größe gemäß den Gleichungen in Abschnitt 3.4 die Messunsicherheit herzuleiten und zu implementieren. So gilt beispielsweise für den Fehler der Konvertereffizienz des ersten Konverters – unter Annahme normalverteilter Fehlerwahrscheinlichkeiten – der in Gleichung 3.13 gegebene Ausdruck.

$$
\begin{aligned}
e_{\eta,\text{c1,S}} &= e_{V_5} + e_{V_7} + e_{R_3} + e_{V_1} + e_{V_2} + e_{R_1} \\
&\overset{3.11}{=} \pm \Bigg[\left(e_{\text{mess},V_5} + \frac{e_{\text{Bereich},V_5} \cdot V_{5,\text{Bereich,max}}}{V_5} \right) \\
&\quad + \left(e_{\text{mess},V_7} + \frac{e_{\text{Bereich},V_7} \cdot V_{7,\text{Bereich,max}}}{V_7} \right) + e_{R_3} \\
&\quad + \left(e_{\text{mess},V_1} + \frac{e_{\text{Bereich},V_1} \cdot V_{1,\text{Bereich,max}}}{V_1} \right) \\
&\quad + \left(e_{\text{mess},V_2} + \frac{e_{\text{Bereich},V_2} \cdot V_{2,\text{Bereich,max}}}{V_2} \right) + e_{R_1} \Bigg] \\
&\overset{3.12}{=} \pm \Bigg[\left(e_{\text{mess},V_5} + \frac{e_{\text{Bereich},V_5} \cdot V_{5,\text{Bereich,max}}}{V_5} \right) \\
&\quad + \left(e_{\text{mess},V_7} + \frac{e_{\text{Bereich},V_7} \cdot V_{7,\text{Bereich,max}}}{V_7} \right) \\
&\quad + I_{R_3}^2 \cdot R_{3,\text{nom}} \cdot R_{3,\text{th}} \cdot \overline{\alpha}_{R_3} + e_{R_3,\text{Fertigung}} \\
&\quad + \left(e_{\text{mess},V_1} + \frac{e_{\text{Bereich},V_1} \cdot V_{1,\text{Bereich,max}}}{V_1} \right) \\
&\quad + \left(e_{\text{mess},V_2} + \frac{e_{\text{Bereich},V_2} \cdot V_{2,\text{Bereich,max}}}{V_2} \right) \\
&\quad + I_{R_1}^2 \cdot R_{1,\text{nom}} \cdot R_{1,\text{th}} \cdot \overline{\alpha}_{R_1} + e_{R_1,\text{Fertigung}} \Bigg]
\end{aligned}
\tag{3.13}
$$

Bei den Testmessungen lagen diese Werte deutlich unter ±0,1%. Setzt man ±0,1% als Unsicherheit ein, so ergibt sich bei einer gemessenen respektive berechneten Effizienz von 95% ein Intervall von [94,905%, 95,095%] für den realen Wert der Effizienz.

3.5.4 Ungenauigkeiten durch Aufbau und Verkabelung

Auch durch die Verkabelung können Ungenauigkeiten in der Messung entstehen, in erster Linie durch die Kabelwiderstände und allfällige parasitäre Induktivitäten.

Zweitere lassen sich durch kurze, möglichst gerade verlegte Kabel sowie die Verdrillung von Hin- und Rückleiter miteinander stark minimieren. Beim Aufbau des Teststandes wurde darauf geachtet, möglichst kurze Kabel ($l < 1$ m) zu verwenden und diese – soweit wie möglich – gerade zu führen und gegeneinander zu verdrillen. Es kann deshalb davon ausgegangen werden, dass die parasitären Induktivitäten vernachlässigt werden dürfen.

Der Widerstand eines Kabels R_K berechnet sich bekanntlich aus Kabellänge l, Leitungsquerschnitt A und spezifischem Widerstand ϱ zu

$$R_\text{K} = \varrho \cdot \frac{l}{A}. \tag{3.14}$$

Für die zur Leistungsführung verwendeten Kupferkabel mit einer Länge von weniger als 1 m und einem Querschnitt von 2,5 mm^2 ergibt sich pro Kabel somit ein Widerstand von $R_\text{K} = 0{,}01786 \, \frac{\Omega\text{mm}^2}{\text{m}} \cdot \frac{1\text{ m}}{2{,}5\text{ mm}^2} \approx 7\text{ m}\Omega$.

Da die Abgriffe der Spannung jedoch unmittelbar am Element stattfinden, hebt sich dieser Effekt für die Konverter-Effizienz auf. Der Einfluss auf die MPP-Effizienz ist vernachlässigbar, da der SAS mittels Sense-Anschlüssen am Konvertereingang immer die programmierte Solarkurve zur Verfügung hält. Der

Spannungsabfall zwischen Konvertereingang und Sense-Anschlüssen des SAS kann dabei vernachlässigt werden, da für den internen Widerstand $R_{\text{SAS,intern}}$ mit Sicherheit $R_{\text{SAS,intern}} \gg R_K$ gilt. Lediglich bei der Gesamteffizienz der Schaltung ist ein gewisser Leistungsverlust zu vermuten. Die Testmessungen haben allerdings gezeigt, dass die theoretische Relation

$$\eta_{\text{tot,c1}} = \eta_{\text{c1}} \cdot \eta_{\text{MPP,c2}} \quad [\%] \tag{3.15}$$

in den meisten Fällen auf drei Nachkommastellen hält. Daraus lässt sich schlussfolgern, dass auch dieser Leistungsverlust kaum Einfluss auf die Messgenauigkeit hat.

Für die Messabgriffe werden Kupferkabel mit einem Leitungsquerschnitt von $1\,\text{mm}^2$ verwendet. Mit einer Kabellänge von < 2 m und Gleichung 3.14 ergibt sich ein Kabelwiderstand von $R_K \approx 36$ mΩ, was gegenüber dem Innenwiderstand der Multimeter von $10\,\text{M}\Omega$ vernachlässigbar ist.

Summa summarum ist also bei dem bestehenden Aufbau eine Vernachlässigung der Ungenauigkeiten durch Aufbau und Verkabelung zulässig, ohne die Messergebnisse dadurch merklich zu verfälschen.

4 Software

4.1 Anforderungen

Die Software des automatischen Teststandes soll die gesamte Steuerung des Teststandes übernehmen sowie eine grafische Oberfläche für die Bedienung durch den Benutzer bieten.

Die Benutzeroberfläche des Teststandes soll dabei verschiedenste Möglichkeiten bieten:

- Konfiguration des SAS (je Kanal): Auswahl von FIX- oder SAS-Modus, Einstellung aller zur Definition des betreffenden Modus' nötigen Parameter.
- Ein- und Ausschalten des SAS (je Kanal).
- Konfiguration der Last: Auswahl des gewünschten Modus (CC, CR, CV oder CP) sowie Definition der nötigen Parameter.
- Ein- und Ausschalten der Last.
- Konfiguration des Triggers: Einerseits Einstellungen betreffend das Triggersignal, andererseits Anzahl der gewünschten Messwerte pro Trigger sowie Dauer der Integration in NPLC.
- Starten einer Einzelmessung oder einer Messreihe.
- Darstellen der Ergebnisse.
- Auswertung der Messergebnisse.
- Speichern der Ergebnisse.
- LiveView-Modus zur Überprüfung der Konfiguration.

Selbstverständlich müssen alle Konfigurationen an die betreffenden Geräte übermittelt werden; die Software muss Messungen starten, Messwerte auslesen und Geräte zurücksetzen können. Des Weiteren muss die Berechnung der nur mathematisch zu ermittelnden Größen nach den Gleichungen 3.5 bis 3.9 sowie die Fehlerberechnung nach Abschnitt 3.5 erfolgen. Zu berücksichtigen ist dabei – insbesondere für die Berechnung der MPP-Effizienz wie auch der Gesamteffizienz –, dass bei der Simulation der Solarkurve gemäß Gleichung 3.1 Fehler bezüglich des MPP auftreten (können).

4.2 Konzept

Während der Einarbeitung in die Bedienung der Messgeräte hat sich erwiesen, dass es möglich ist, die Messgeräte direkt aus MATLAB zu steuern. Da die Datenaufbereitung sowieso in MATLAB erfolgen soll, wurde deshalb entschieden, LABVIEW als zusätzliche Instanz aus dem Messaufbau zu entfernen und die Fernsteuerung der Messgeräte direkt in MATLAB zu implementieren.

Die Software an sich wird möglichst modular aufgebaut, ähnliche Aufgaben werden in einzelnen, gerätespezifischen Funktionen implementiert. Ziel dieser Überlegung ist es, nach Implementation der Basisfunktionen komplexere Aufgaben schnell, einfach und sicher mit Hilfe der bereits bestehenden Funktionen zu implementieren.

In allen Funktionen, die direkt auf die Geräte zugreifen, muss sichergestellt werden, dass eventuell auftretende Fehler direkt festgestellt und das betroffene Gerät zurückgesetzt und neu initialisiert wird. In bestimmten Fehlerfällen wird – falls es die Konstellation nötig macht – die gesamte Teststandes-Steuerungssoftware neu gestartet.

Als weiteres Konzept kommt die Ausgabe von Informationen an den Benutzer mittels MATLAB-Befehlsfenster zum Einsatz. Es werden Informationen, Warnungen und Fehler unterschieden, wobei letztere zu einem Neustart des Systems führen. Sämtliche Meldungen, die (mit Zeitstempel) im Befehlsfenster angezeigt werden, werden auch in eine Log-Datei geschrieben. Somit können eventuelle Fehler später rekonstruiert oder Messungen einfacher wiederholt werden.

4.3 Grundlagen der Programmierung

Zur Erstellung der Benutzeroberfläche in MATLAB wird der MATLAB-eigene GUI-Editor `guide` verwendet. Dieser Editor ermöglicht die grafische Gestaltung der Oberfläche und erstellt automatisch eine .m-Datei, welche die nötigen Funktionssignaturen bereits enthält. Es genügt also, die Oberfläche zu gestalten und später in der Quelltext-Vorlage die Funktionalität zu implementieren.

Um die Geräte aus MATLAB zu steuern, muss die Instrument Control Toolbox vorhanden sind. Diese unterstützt die Übermittlung von Befehlen an SCPI-Geräte mittels `visa` oder `gpib`. Zu diesem Zweck muss zuerst ein Verbindungsobjekt erstellt werden, im Falle einer visa-Verbindung lautet der benötigte Befehl

```
VerbObj = visa('Hersteller', 'Adresse'),
```

wobei der Hersteller in diesem Falle `agilent` ist; die Adresse setzt sich zusammen als

```
USB::HerstID::DevID::SERIAL::INSTR.
```

Dabei ist HerstID die Identifikationsnummer des Herstellers, DevID die Identifikationsnummer des Gerätemodells und SERIAL die Seriennummer des einzelnen Gerätes. Die jeweiligen Werte sind in den gerätebezogenen Sektionen des Abschnitts 3.2 angegeben.

Für die Herstellung einer Verbindung zu einem GPIB-Gerät lautet der nötige Befehl

```
VerbObj = gpib('Hersteller', BoardAddress, PrimaryAddress).
```

Diese Art der Verbindung ist nur für die elektronische Last erforderlich, die nötigen Daten sind in Abschnitt 3.2.6 angegeben; der Hersteller des verwendeten GPIB-Adapters ist, wie bereits erwähnt, National Instruments (ni).

Für die benötigten SCPI-Befehle kann in Handbüchern oder Referenzen nachgeschlagen werden.[19]

In MATLAB muss nach Herstellen der Verbindung (s. o.) diese mit dem Befehl `fopen(VerbObj)` geöffnet werden, bevor mit den Befehlen `fprintf(VerbObj, 'Befehlsstring')` und `ans = query(VerbObj, 'Anfrage')` Befehle gesendet respektive Werte ausgelesen werden können. Mittels `fclose(VerbObj)` kann die Verbindung geschlossen werden.

[19] Diese sind meist Teil der Bedienungsanleitung des betreffenden Gerätes.

4.4 Funktionalität

4.4.1 Einführung

In diesem Abschnitt soll die Funktionalität der Software in Form einer kurzen Bedienungsanleitung dargestellt werden. Dabei wird jeder Funktionalität ein Abschnitt gewidmet. Um die Anwendung der Software repektive eine spätere Weiterentwicklung auch nicht-deutschsprachigen Personen zu ermöglichen, wurden Software und Oberfläche in Englisch programmiert. In dieser Beschreibung werden die Bezeichnungen so verwendet, wie sie auch auf der Benutzeroberfläche verwendet werden; es erfolgt keine Übersetzung der Bezeichnungen, insbesondere von Schaltflächennamen, ins Deutsche. **Abbildung 4.1** zeigt die Oberfläche vor dem Starten des Systems.

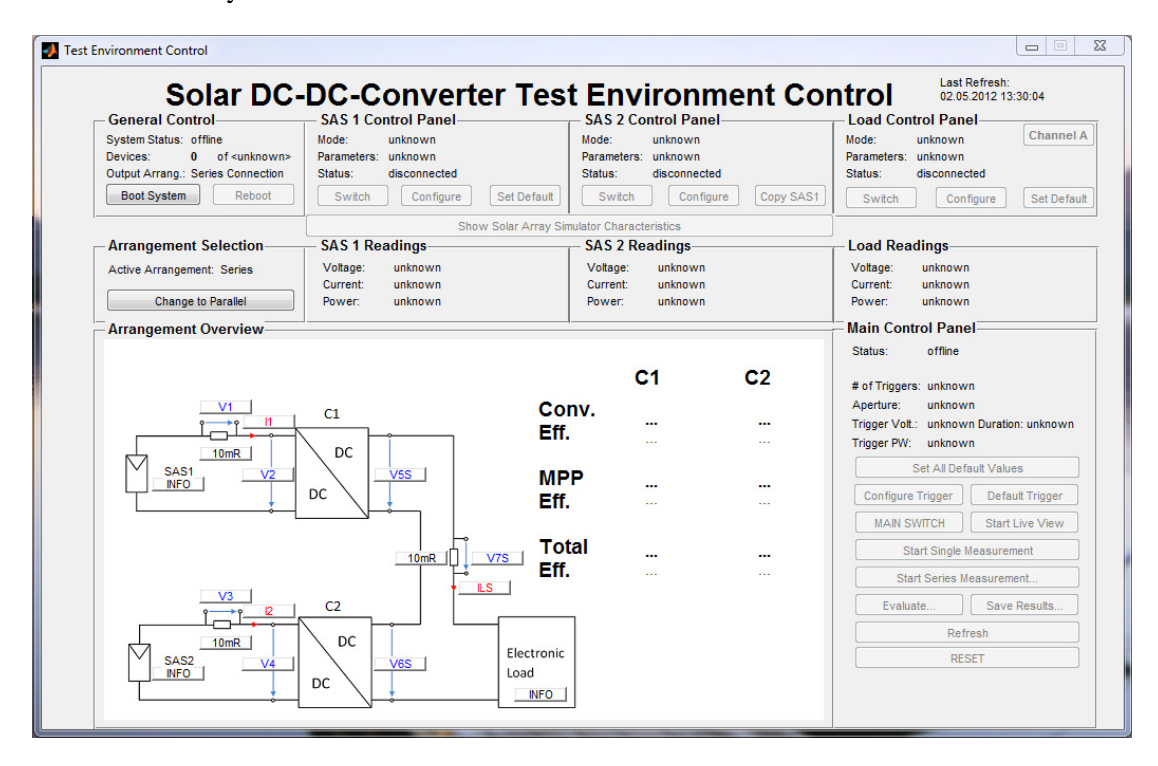


Abbildung 4.1: *Hauptoberfläche des Teststandes vor dem Starten des Systems*

4.4.2 Start der Oberfläche

Die Software besteht aus einem Verzeichnis, das mehrere Unterordner (mit Funktionen) enthält und in dem die Datei `Start.m` liegt. Um die Benutzeroberfläche zu starten, muss zu diesem Dateipfad in MATLAB navigiert werden. Durch die Eingabe des Befehls `Start` in das MATLAB-Befehlsfenster wird die Oberfläche gestartet. Die benötigten Unterordner werden automatisch zur in MATLAB bekannten Ordnerhierarchie hinzugefügt.

Nach dem Laden der Oberfläche sind alle Schaltflächen – mit Ausnahme der Schaltflächen *Boot System*, *Change to Parallel* sowie der Element-Schaltflächen in der Schaltungsübersicht – deaktiviert. Grundsätzlich werden immer dann Schaltflächen deaktiviert, wenn ihre Funktionalität nicht verfügbar ist *oder* ihre Benutzung die Systemstabilität gefährden würde.

4.4.3 Auswahl der Schaltung

In Abschnitt 3.1 wurden zwei zu unterstützende Schaltungsarten vorgestellt: Serienschaltung und Parallelschaltung der Konverterausgänge. Standardmäßig ist die Serienschaltung ausgewählt, mit Hilfe der Schaltfläche *Change to Parallel* kann auf Parallelschaltung umgestellt werden.[20] Diese Auswahl muss *vor dem Starten des Systems geschehen und mit der tatsächlich aufgebauten Schaltung übereinstimmen* (vgl. **Abbildung 3.1**). Ist die falsche Schaltungsart ausgewählt, so wird die automatische Auswertung beliebig falsche Ergebnisse liefern.

4.4.4 Starten, Herunterfahren und Neustarten des Systems

Mittels der Schaltfläche *Boot System* wird das Messsystem gestartet. Dies bedeutet, dass die Software versucht, zu jedem der Geräte eine Verbindung herzustellen, jedes Gerät zurückzusetzen und dessen Standardwerte zu laden. Im Feld *General Control* wird der derzeitige Systemstatus angezeigt, die Anzahl der verbundenen Geräte sowie die aktive Schaltungsart.

Nach einem erfolgreichen Systemstart sind alle Schaltflächen – mit Ausnahme der Schaltflächen *Evaluate...*, *Save Results...* sowie der Auswahl der Schaltungsart – aktiviert. Die Schaltfläche *Boot System* heißt nun *Suspend System*; die Verwendung der Schaltfläche *Reboot* ist gleichbedeutend mit der Verwendung der Schaltfläche *Suspend System* und anschließender Verwendung von *Boot System*.

Ein Neustart des Systems kann dann sinnvoll sein, wenn entweder ein Fehler an einem Gerät aufgetreten ist, so dass dieses neu gestartet werden musste, oder wenn Änderungen im Quelltext der Software vorgenommen wurden.

Wichtig: Sollen ein oder mehrere Geräte abgeschaltet oder abgetrennt werden, *muss* das System zuvor mittels *Suspend System* heruntergefahren werden! Anderenfalls wird MATLAB nicht in der Lage sein, die Verbindung zu diesem Gerät wiederherzustellen; MATLAB muss neu gestartet werden.

Die Oberfläche kann jedoch bedenkenlos mittels des weißen Kreuzes auf rotem Grund geschlossen werden; in diesem Falle wird die Routine, die das System herunterfährt, automatisch ausgeführt.

4.4.5 Steuerung des Solar Array Simulators

Jeder Ausgang des SAS verfügt über ein eigenes Kontroll- sowie Lesewertfeld, gekennzeichnet mit *SAS 1 Control Panel* und *SAS 1 Readings* respektive *SAS 2 Control Panel* und *SAS 2 Readings*. Die Steuerung der Simulatoren erfolgt jeweils über drei Schaltflächen (v. l. n. r.):

- An-Aus-Schalter *Switch ON / Switch OFF*,
- Einstellungs-Schalter *Configure* und
- Zurücksetzen auf Standards (*Set Default*) für SAS_1 respektive Übernahme der Einstellungen von SAS_1 für SAS_2 (*Copy SAS1*).

Die Verwendung des **Schalter**s aktiviert dabei den im oberen Teil des Kontrollfeldes angegebenen Modus mit seinen ebenfalls angezeigten Parametern. Das Feld *Status* gibt Auskunft über den aktuellen Zustand des SAS-Ausgangs (eingeschaltet, ausgeschaltet, unbekannt). Ist ein Modul eingeschaltet, so wird der Hintergrund des betreffenden Kontrollfeldes orange eingefärbt. *Ist der Hintergrund eines Kontrollfeldes rot eingefärbt, sollte die Arbeit unterbrochen und der Zustand des SAS direkt am Gerät geprüft werden.*

Die **Konfiguration** wird durch einen Klick auf die Schaltfläche *Configure* geöffnet (**Abbildung 4.2**). Sie ermöglicht die Auswahl der Betriebsmodi FIX und SAS, die in Abschnitt 3.2.2 beschrieben sind, sowie

[20] Ist Parallelschaltung ausgewählt, so trägt die Schaltfläche die Aufschrift *Change to Series*.

die Konfiguration der jeweiligen Parameter. Bei Verwendung der Schaltfläche *Apply All Settings* wird zuerst getestet, ob die eingegebenen Werte zulässig sind. Gültige Werte werden an den SAS gesendet, bei ungültigen Werten erscheint eine Warnung in derselben Benutzeroberfläche.

Wichtig: Wird die Schaltfläche zum Umschalten des aktuellen Modus verwendet, so werden alle Konfigurationsparameter auf die derzeit eingestellten Werte zurückgesetzt! Wird das Fenster mittels der Schließen-Schaltfläche oben rechts geschlossen, werden keine Änderungen an der bestehenden Konfiguration vorgenommen.

Die **Lesewerte** des Gerätes werden jeweils so aktuell wie möglich angezeigt. Da diese Werte jedoch bei keiner Berechnung zum Einsatz kommen, kann über ihre Genauigkeit keine Aussage getätigt werden; sie dienen lediglich dazu, einen Eindruck über den Zustand des Systems zu gewinnen.

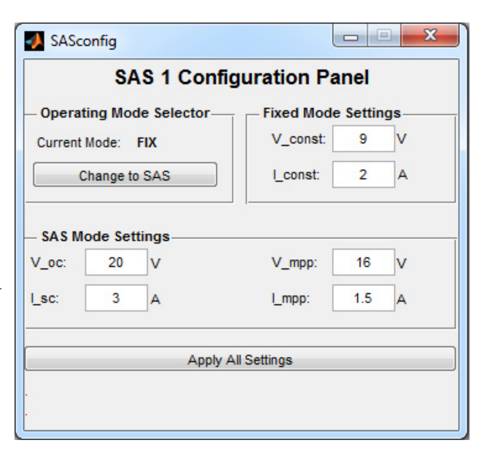


Abbildung 4.2: *SAS-Konfigurationsdialog*

4.4.6 Anzeige der Solarkurven

In Abschnitt 3.2.2 wurde darauf hingewiesen, dass das für die Berechnung der Solarkurve verwendete Modell zwar stets für die Werte V_{oc} und I_{sc} exakt ist, die Werte $V_{mpp,real}$ und $I_{mpp,real}$ (und somit auch $P_{mpp,real}$) von den eingestellten Werten $V_{mpp,def}$, $I_{mpp,def}$ und $P_{mpp,def}$ abweichen können. Die Software des Teststandes wertet die Modellgleichung 3.1 intern aus und führt alle Effizienz-Berechnungen auf Grund der berechneten, realen Werte durch.

Zu Informationszwecken wurde die Schaltfläche *Show Solar Array Simulator Characteristics* zwischen den Kontroll- und Lesewertfeldern einfügt. Diese Schaltfläche bewirkt die Erzeugung einer Grafik, in der die simulierte Solarkuve als $V(I)$ dargestellt wird. Auf dieser Kurve sind der definierte und der reale MPP eingezeichnet, die Werte $P_{mpp,def}$ und $P_{mpp,real}$ sind in der Legende angegeben. In den **Abbildungen 3.4(a)** und **3.4(b)** (S. 16 und 16) sind solche Grafiken dargestellt.

4.4.7 Steuerung der Last

Die Steuerung der Last untscheidet sich nur unwesentlich von der Steuerung des SAS: Auch hier werden aktueller Modus, eingestellte Parameter und gegenwärtiger Status im Kontrollfeld *Load Control Panel* dargestellt; die gegenwärtigen Werte von Spannung, Strom und Leistung werden im Lesewertfeld *Load Readings* angezeigt. Die Steuerung der Last erfolt über vier Schaltflächen:

- An-Aus-Schalter, exakt wie beim SAS,
- Einstellungs-Schalter *Configure*,
- Zurücksetzen auf Standards (*Set Default*), exakt wie beim SAS, und
- Kanalwahlschalter (*Channel A* / *Channel B*).

Das **Konfigurationsfenster** der Last (**Abbildung 4.3**) unterscheidet sich in Aussehen und Funktionalität etwas von dem des SAS.

Mit der Einstellung *Operation Mode* kann der aktuell gewünschte Modus eingestellt werden. Typisch für Gleichspannungswandler mit Anwendung im Solarbereich ist die Einstellung *Constant Voltage (CV)*

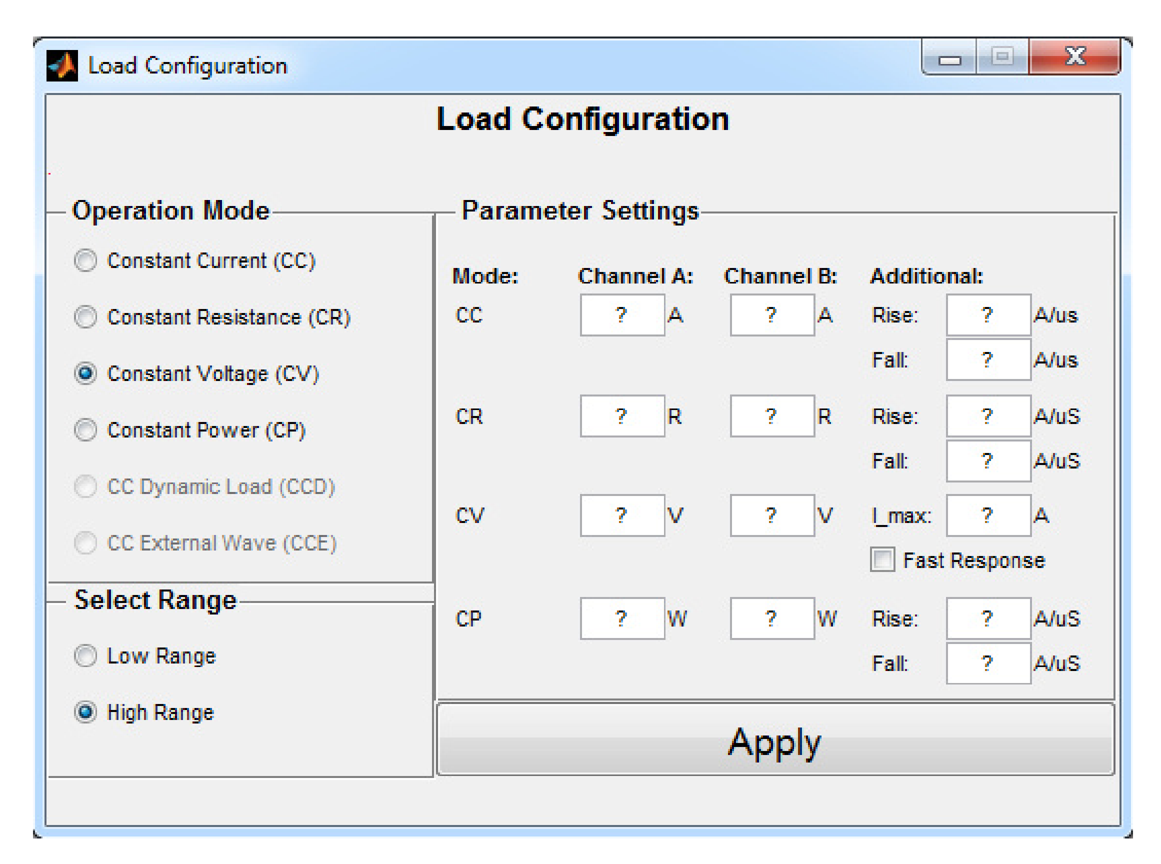


Abbildung 4.3: *Last-Konfigurationsfenster. Die Abbildung zeigt eine Offline-Version des Konfigurationsfensters; im Betrieb würden die '?' durch die aktuell eingestellten Werte ersetzt.*

(= Konstantspannung), aber auch andere Modi können gewählt werden. In dieser Version der Software werden die Modi *Constant Current Dynamic Load* und *Constant Current External Wave* nicht unterstützt.

Das Kontrollfeld *Select Range* ermöglicht die Auswahl *Low Range* oder *High Range*, wobei der Unterschied zwischen den beiden Modi in [Chroma HB, p. 3-8] wie folgt beschrieben ist: „The low range is used for input voltage in low voltage range while the high range is for the input voltage over low voltage range.“ Zu deutsch: Low-Range wird für Spannungen im Low-Range-Bereich verwendet, High-Range für Spannungen oberhalb des Low-Range-Spannungsbereichs... Auf p. 4-13 wird diesbezüglich angemerkt, dass die Range-Einstellung einen Einfluss auf die Auflösung hat; mit der Einstellung *Low Range* kann im unteren Spannungsbereich eine höhere Auflösung erreicht werden. Wie der untere Spannungsbereich definiert ist, geht aus [Chroma HB] jedoch nicht hervor.

Das Kontrollfeld *Parameter Settings* enthält alle möglichen Parameter für alle unterstützten Modi und zeigt, wenn das Fenster geöffnet wird, die aktuell eingestellten Werte an. *Rise* und *Fall* geben jeweils an, mit welcher Geschwindigkeit der Strom repektive der Widerstand seinen Wert ändern darf (in A/μs respektive A/μS). Für den Konstantspannungsbetrieb (CV) werden stattdessen der größte zulässige Strom $I_{\max}$ sowie die Reaktionsgeschwindigkeit (= Anstiegsrate des Stroms) der Last (schnell / langsam) angegeben.

Im Unterschied zum SAS löst ein Klick auf die Schaltfläche *Apply* nur eine *Überprüfung jener Werte aus, die für den ausgewählten Modus von Bedeutung sind*; auch werden nur die *Werte für den gewählten Modus* an die Last gesendet. Wird das Konfigurationsfenster mittels der Schaltfläche oben rechts geschlossen, so wird keiner der eingestellten Werte an die Last gesendet.

Der erwähnte **Kanalwahlschalter** implementiert die von der Last unterstützten zwei verschiedenen Kanäle, die unterschiedlich konfiguriert werden können. Es kann somit für Kanal A z. B. eine andere Spannung vorgegeben werden als für Kanal B. Mit dem Kanalwahlschalter im Lastkontrollfeld kann zwischen den beiden Kanälen umgeschaltet werden; auf der Schaltfläche wird jeweils der derzeit aktive Kanal angezeigt.

Das **Lesewertfeld** dient wie beim SAS lediglich der Information, da die Messgenauigkeit bei Weitem nicht an jene der Tischmultimeter heranreicht.

4.4.8 LiveView

Der Modus *LiveView*, zu starten durch die Schaltfläche *Start Live View*, konfiguriert den Teststand so, dass etwa einmal pro Sekunde alle Messwerte ausgelesen, Ströme und Effizienzen berechnet und angezeigt werden. Durch erneutes Betätigen derselben Schaltfläche, die bei aktivem LiveView die Beschriftung *Stop Live View* trägt, werden die Standardeinstellungen für den Messstand wiederhergestellt und die regelmäßige Aktualisierung beendet.

Man beachte, dass sowohl das Starten als auch das Beenden des LiveView-Modus einige Sekunden in Anspruch nehmen, da jeweils alle Multimeter sowie der Funktionsgenerator umkonfiguriert werden (müssen). Im *Main Control Panel* wird unter dem Punkt *Status* der aktuelle Zustand des Systems angezeigt; für den LiveView-Modus sind *Preparing Live View...*, *Live View* und *Stopping Live View* vorgesehen. Es ist empfehlenswert, keine Aktionen vorzunehmen, solange einer dieser Modi aktiv ist.

4.4.9 Durchführung von Messungen

Messung einrichten

Mittels der Schaltfläche *Configure Trigger* wird ein Konfigurationsfenster geöffnet, in dem die Einstellungen für den Trigger gesetzt werden können (**Abbildung 4.4**).

Number of Measurements gibt an, aus wievielen Teilmessungen die Ergebnisse berechnet werden sollen. Zulässig sind Werte von 1 bis 50 000; die Standardeinstellung sieht 10 000 Teilmessungen vor. *Aperture* gibt an, wie lange das Eingangssignal integriert werden soll (in NPLC). *Trigger Voltage* gibt die Spannung des Triggersignals vor, sie sollte gemäß Multimeter-Eigenschaften 3 V bis 5 V betragen (Abschnitt 3.2.3); *Trigger Pulsewith* gibt die Breite des Pulses vor, die 1 μs bis 2 μs betragen sollte.

Beim Klicken auf *Apply* werden die Werte auf ihre Gültigkeit hin überprüft. Sind alle Werte gültig, so werden sie an den Funktionsgenerator und die Multimeter übermittelt; andernfalls erscheint im Konfigurationsfenster selbst eine Warnung. Das Schließen des Fensters mittels des weißen Kreuzes auf rotem Grund verwirft alle vorgenommenen Änderungen.

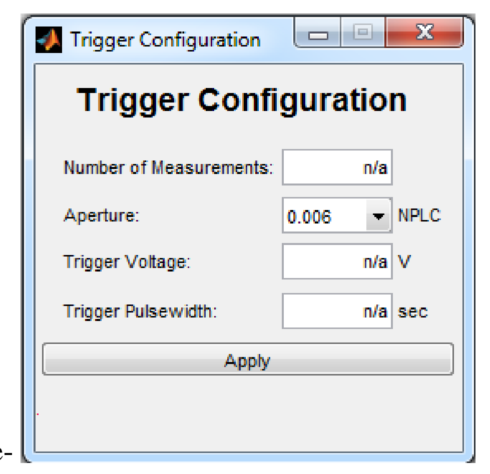


Abbildung 4.4: *Fenster zur Messungskonfiguration (Trigger-Konfiguration)*

Die Schaltfläche *Default Trigger* stellt die Standard-Triggereinstellungen wieder her, namentlich 10 000 Messungen mit einer Integrationszeit von 0,006 NPLC; Triggerspannung 5 V und Pulsbreite 1,5 μs.

Einzelmessung starten

Nachdem die Messung mittels Triggereinstellungen wunschgemäß konfiguriert wurde, kann sie durch einen Klick auf die Schaltfläche *Start Single Measurement* gestartet werden. Das System deaktiviert bis zum Abschluss der Messung (nahezu) alle Schaltflächen. Während einer Messung durchläuft das System folgende Zustände: *Starting Single Measurement...*, *Measuring* und *Collecting Data...*. Nach Ende der Messung werden die Ergebnisse im Befehlsfenster von MATLAB angezeigt. Mittels der Schaltfläche *Save Results...* können die Ergebnisse in eine mat-Datei exportiert werden, die Schaltfläche *Evaluate...* generiert Graphen zur Auswertung der Messung (siehe Abschnitt 4.4.10).

Die Ergebnisse einer Einzelmessung enthalten jeweils für die Spannungen V_1 bis V_7 nebst den Mittelwerten und Fehlern gemäß Abschnitt 3.5 *alle* einzelnen Messwerte (also bis zu 50 000) sowie eine von MATLAB berechnete Standardabweichung aller Messwerte.

Messreihe starten

Auch für eine Messreihe müssen die Triggereinstellungen – wie vorangehend beschrieben – gesetzt werden. In diesem Falle beziehen sich die Einstellungen auf eine einzelne Messung der Messreihe. Als Messreihe kommt theoretisch eine Variation der Last, eine Variation eines oder beider Ausgänge des SAS sowie jede beliebige Kombination davon in Frage. Aus Zeitgründen musste bei der Implementation der Software leider auf jegliche Variationsmöglichkeit für die SAS-Ausgänge verzichtet werden, eine spätere Implementation ist allerdings vorgesehen.

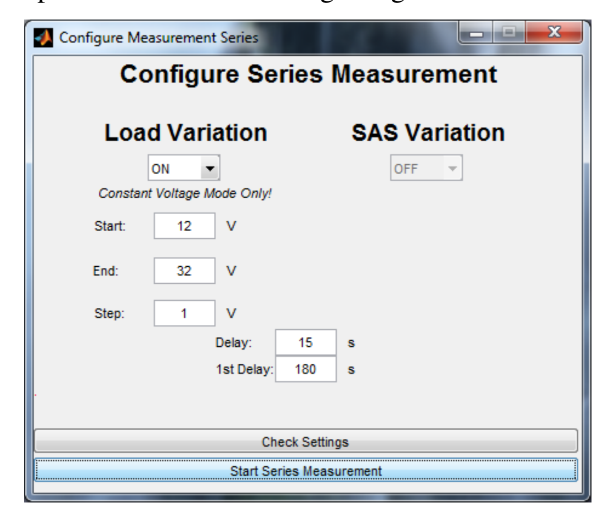


Abbildung 4.5: *Konfigurationsfenster für Messreihen*

Um eine Messreihe zu starten, steht die Schaltfläche *Start Series Measurement...* zur Verfügung. Es öffnet sich ein Dialogfenster (**Abbildung 4.5**), in dem die gewünschten Einstellungen vorgenommen werden können. Auf der linken Seite sind die Einstellungen für die Lastvariation zu tätigen, die rechte Seite ist für eine spätere Implementation der SAS-Variation vorgesehen.

Um die **Lastvariation** zu aktivieren, muss in der Auswahlbox unter dem Titel *Load Variation* der Eintrag *ON* gewählt werden. Da die Lastvariation (bisher) nur für den Konstantspannungsbetrieb vorgesehen ist, sind des Weiteren Start- und Endspannung sowie die Schrittweite anzugeben. Dabei werden Start- und Endwert als ausschlaggebend betrachtet, das Vorzeichen der Schrittweite (*Step*) wird automatisch angepasst.

Während der Testmessungen wurde festgestellt, dass einige Gleichspannungswandler eine relativ lange Zeit (etwa 2 Minuten) benötigen, um sich auf geänderte Bedingungen – sei es am Eingang oder am Ausgang – einzustellen.[21] Deshalb wurden die Einstellmöglichkeiten *Delay* und *1st Delay* geschaffen: *Delay* bestimmt, wie lange (in Sekunden) zwischen zwei Messungen einer Reihe – nach Änderung der Parameter – gewartet werden soll; *1st Delay* bestimmt, wie lange nach Einschalten des Systems gewartet werden soll, bevor die erste Messung beginnt. Die in **Abbildung 4.5** eingetragenen Werte haben sich als gut praktikabel erwiesen.

[21] Die Einstellung dauert umso länger, umso größer die Änderung ist. Deshalb wird darauf verzichtet, die betroffenen Geräte zur Änderung der Parameter explizit auszuschalten.

Mittels der Schaltfläche *Check Settings* kann überprüft werden, ob die Einstellungen gültig sind. Dieser Test wird allerdings ebenfalls bei Auswahl der Schaltfläche *Start Series Measurement* ausgeführt; sind die Werte ungültig, wird eine Warnung ausgegeben und die Messung *nicht* gestartet (das Konfigurationsfenster bleibt geöffnet). Sobald jedoch die Schaltfläche *Start Series Measurement* bei gültigen Werten angeklickt wird, startet die Messreihe sofort.

Bei einer Messreihe sind verschiedene Status-Meldungen im *Main Control Panel* vorgesehen: *Starting Series Measurement...*, *Setting new load params...* (Anpassung der Lastspannung), *Waiting for Converter* (während des Delays), *Initializing Measurement # of ##* (Einstellen der Geräte), *Measurement # of ##* (Messung läuft) und *Collecting Data for Run # of ##* (Lesen der Messwerte). Dabei symbolisiert # jeweils die Nummer der aktuellen Messung und ## die Gesamtanzahl durchzuführender Messungen.

Wichtig: Stehen beide der verfügbaren Auswahllisten auf OFF, so wird automatisch eine Einzelmessung gestartet. Um das Starten einer Messreihe abzubrechen, steht wiederum das übliche Schließen des Fensters zur Verfügung.

Nach Abschluss einer Teilmessung werden die Ergebnisse derselben jeweils angezeigt, nach Abschluss der Messreihe wird automatisch die Auswertung (entspricht einem Klick auf die Schaltfläche *Evaluate...*) ausgeführt. Bei Messreihen werden jeweils nur die Mittelwerte der in den Teilmessungen aufgenommenen Spannungen in der Ergebnisvariablen abgelegt.

Die Messreihe dient in erster Linie dazu, das Verhalten der Konverter bei variablen Bedingungen zu testen. Ein optimaler Konverter würde – unabhängig von der Lastspannung – immer im MPP laufen und so die maximale Leistung liefern. Je größer die Abweichung des Konverters vom MPP, umso größer ist auch das Verbesserungspotential.

4.4.10 Auswerten und Speichern von Messungen

Ein Klick auf die Schaltfläche *Evaluate...* startet eine Routine, die – abhängig vom Typ der letzen Messung – Graphen zur Auswertung der Ergebnisse erstellt. Bei der Auswertung der Messung kommt der reale (mittels Gleichung 3.1 bestimmte) MPP zur Anwendung.

Für eine **Einzelmessung** werden in der ersten von zwei Grafiken alle Spannungen über der Zeit aufgetragen. Dabei werden jeweils die Ein- und Ausgangsspannung eines Gleichspannungswandlers in einem Diagramm sowie alle Spannungen, die zur Stromberechnung dienen, in einem Diagramm dargestellt; die erste Grafik enthält somit drei Graphen. Anhand der Eingangsspannungen der Konverter lässt sich dann z. B. der MPP-Tracking-Algorithmus erkennen. Mit Hilfe der Ausgangsspannung lässt sich erkennen, ob der Konverter im Hoch- oder Tiefsetzstellerbetrieb arbeitet oder gar einfach durchschaltet.

Die zweite Grafik enthält je Gleichspannungswandler zwei Graphen, insgesamt also vier. Für jeden der Konverter wird die Solarkurve gemäß Gleichung 3.1 als $I(V)$ dargestellt. Darin eingezeichnet ist der (reale) MPP sowie alle Messpunkte von Strom und Spannung am Konvertereingang. Dieser Graph ermöglicht eine Aussage über die Genauigkeit des MPP-Tracking-Algorithmus: Je näher alle Messwerte des Konvertereingangs am realen MPP liegen, desto besser ist der Algorithmus.
Der zweite Graph der zweiten Grafik stellt die Leistung am Konvertereingang gegenüber der Spannung am Konvertereingang dar. Auch dieser Graph ermöglicht eine Aussage über den MPP-Tracking-Algorithmus: Je mehr Messpunkte bei der (realen) MPP-Spannung und somit bei der höchstmöglichen Leistung liegen, desto besser ist der MPP-Tracking-Algorithmus.

Für eine **Messreihe** des Typs ***L-Series***[22] werden ebenfalls zwei Grafiken erstellt. In der ersten werden für jeden Konverter drei Graphen erstellt: Die Konvertereffizienzen (Konverter, MPP und gesamte), die Konverterspannungen (Eingang, Ausgang und MPP-Spannung) sowie die Konverterströme (Eingang, Ausgang und MPP-Strom), jeweils gegenüber der Lastspannung. Da die Fehler automatisch zu jeder Messung berechnet und abgelegt werden, können direkt sogenannte *Fehlerbalken* (*errorbars*) erzeugt werden, die angeben, in welchem Bereich der Wert unter Berücksichtigung des zu erwartenden Fehlers liegt.

In der zweiten Grafik werden für jeden Konverter zwei Graphen erzeugt: Einer, der die Konverterleistungen (Eingang, Ausgang und MPP-Leistung) gegenüber der Lastspannung darstellt und ein zweiter, der die drei Effizienzen gegenüber der Konverterausgangsspannung darstellt. Auch in diesem Falle werden wieder Fehlerbalken verwendet, die der Darstellung der Messunsicherheit dienen.

In Kapitel 5 sind einige Beispiele solcher Auswertungsgrafiken abgebildet. Die direkt in MATLAB erzeugten Figuren (*figures*) können in verschiedenste Grafik-Formate exportiert und somit gespeichert werden. Um auch die Werte zu speichern, steht die Schaltfläche *Save Results...* zur Verfügung. Sie öffnet einen Dialog, der zur Auswahl eines Dateipfades und zur Angabe eines Namens auffordert, unter dem anschließend die Variable `result` (vgl. Abschnitt A.3, S. 57) als mat-Datei gespeichert wird.

4.4.11 Abbruch einer Messung

Der Abbruch von Messungen ist grundsätzlich *nicht vorgesehen*, da keine Möglichkeit gefunden werden konnte, alle betroffenen, laufenden Prozesse in MATLAB zu beenden. Soll eine Messung – aus welchen Gründen auch immer – dennoch abgebrochen werden, so ist dies durch das Schließen der Benutzeroberfläche mittels des weißen Kreuzes auf rotem Grund möglich. Dadurch wird das gesamte System heruntergefahren und muss erneut durch Eingabe des Befehls `Start` in das Befehlsfenster gestartet werden.

4.4.12 Systemweite Funktionen in der Hauptkontrolle

Im Hauptkontrollfeld (*Main Control Panel*) sind einige Funktionen implementiert, die das gesamte Testsystem betreffen. Diese sind im Folgenden kurz erläutert:

Set All Default Values setzt für alle Geräte – Multimeter, Funktionsgenerator, SAS und elektronische Last – die Standardeinstellungen, die in verschiedenen Dateien im Ordner *settings* definiert sind.

Der *MAIN SWITCH* (Hauptschalter) kann nur betätigt werden, wenn sowohl SAS als auch elektronische Last mit dem Computersystem verbunden sind. Sind beide SAS-Ausgänge wie auch die Last ausgeschaltet, bietet dieser Schalter – in diesem Fall mit *ALL ON* beschriftet – die Möglichkeit, mit einem Klick die drei genannten Geräte einzuschalten. Ist mindestens eines der Geräte eingeschaltet, so werden mit dem – in diesem Falle mit *ALL OFF* beschrifteten – Schalter alle eingeschalteten Geräte ausgeschaltet.

Die Schaltfläche *Refresh* dient dazu, die gesamte Benutzeroberfläche neu zu laden, um die aktuellsten Werte und Zustände anzuzeigen.

Mit der Schaltfläche *RESET DEVICES* wird das Skript `SystemReset.m` (siehe Anhang A.4, S. 75) aufgerufen, welches an alle Geräte RESET-Befehle schickt. Diese Funktion kommt in erster Linie dann zum Einsatz, wenn bei irgendeinem Gerät ein Fehler („ERROR“) aufgetreten ist.

[22] *L-Series* ist die programminterne Bezeichnung einer Messreihe mit (ausschließlicher) Lastvariation.

4.4.13 Anzeige der Ergebnisse

Das Feld *Arrangement Overview* enthält ein Schaltbild der aktuell aktiven Schaltung (Serienschaltung oder Parallelschaltung), in der alle Geräte und Messwerte eingezeichnet sind. Für SAS_1, SAS_2 sowie die elektronische Last existiert eine Schaltfläche *Info*, die Widerstandswerte sind ebenfalls je auf einer Schaltfläche angegeben. Auch die Bezeichnungen der Spannungen und Ströme vor dem Starten der Messung (siehe **Abbildung 4.1**) respektive ihre gemessenen Werte nach einer Messung (**Abbildung 4.7** zeigt die GUI nach einer Einzelmessung bei Serienschaltung) werden auf einer Schaltfläche angezeigt.

Diese Schaltflächen dienen dazu, weitere Informationen über die betreffende Größe zu erhalten, so z. B. Informationen zwecks Identifikation über den Gerätetyp, die Seriennummer, die Adresse, den Verbindungstyp und die Instituts-Inventarnummer (RegNo.), aber auch messtechnische relevante Informationen wie den eingestellten Messbereich, den exakten Wert und den berechneten Messfehler. **Abbildung 4.6** zeigt ein solches „Device Information"-Fenster für den Widerstand R_1.

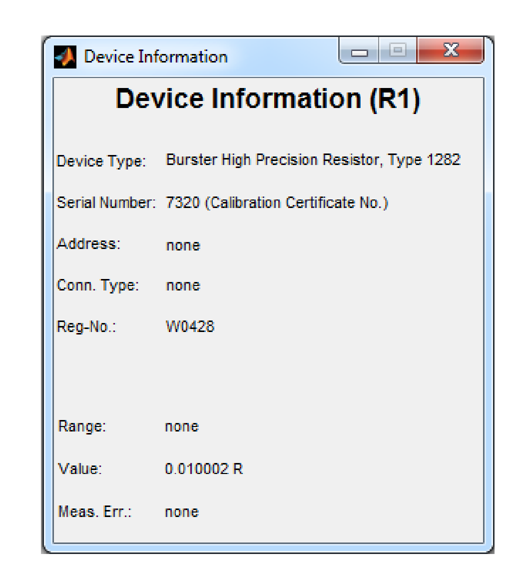


Abbildung 4.6: *„Device Information"-Fenster für den Widerstand R_1*

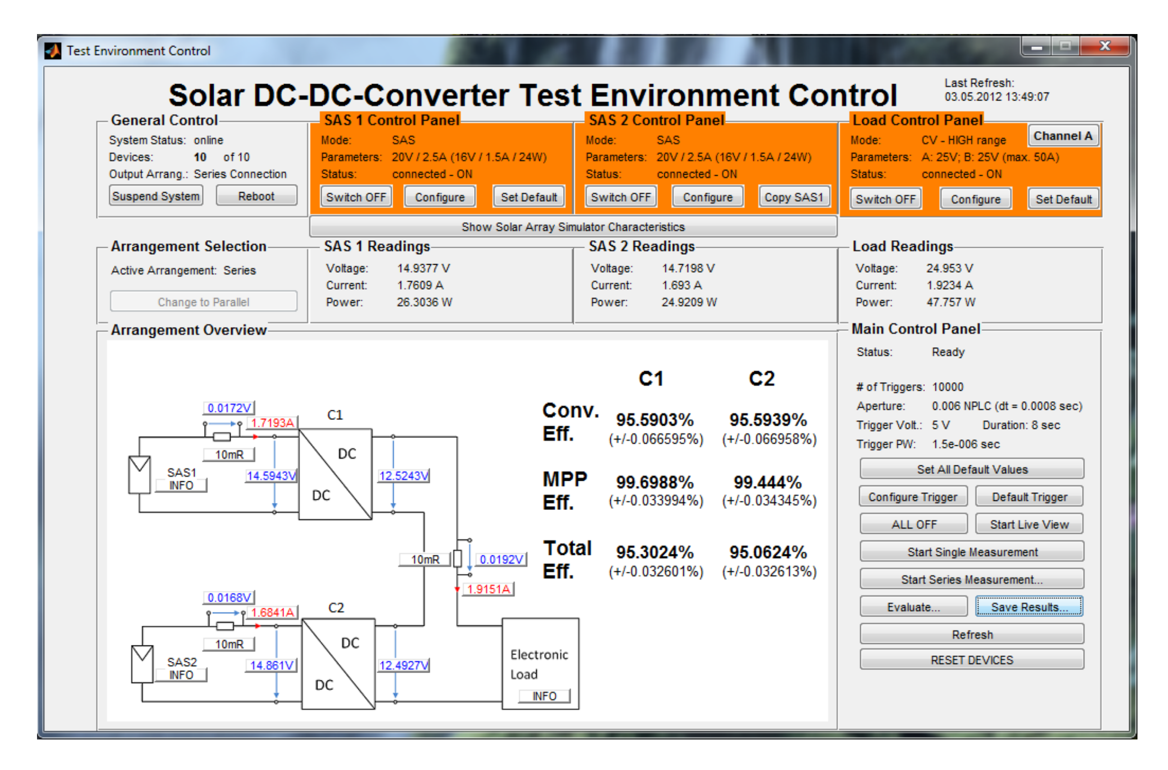


Abbildung 4.7: *Benutzeroberfläche mit Anzeige des Ergebnisses einer Einzelmessung (Serienschaltung). Der orange Hintergrund der drei Kontrollfelder zeigt an, dass beide Ausgänge des SAS wie auch die Last eingeschaltet sind.*

5 Ergebnisse der Testmessungen

5.1 Allgemeine Feststellungen

Die hier beschriebenen Testmessungen wurden allesamt mit dem Evaluations-Kit des Gleichspannungswandlers *SolarMagic SM3320-1A1* durchgeführt (Daten siehe **Tabelle 2.1**). Einige (generelle) Ergebnisse der Testmessungen wurden bereits vorgängig erwähnt; an dieser Stelle soll noch einmal auf sie eingegangen werden:

Werden Änderungen an der Systemkonfiguration vorgenommen – beispielsweise eine Änderung der Konverterein- oder -ausgangsspannung –, so benötigen die Konverter eine nicht zu vernachlässigende Zeit, um sich auf die neue Situation einzustellen. Insbesondere der MPP-Tracking-Algorithmus benötigt nahezu 2 Minuten, um auf die neue Situation anzusprechen und anschließend noch wenige Sekunden, um den MPP tatsächlich zu finden. Des Weiteren wurde eine gewisse Proportionalität zwischen der Intensität der Änderung und der Dauer, bis der Konverter im MPP arbeitet, festgestellt: Je größer die Änderung (z. B. der Lastspannung) ist, desto länger braucht der Konverter, sich auf die neue Situation einzustellen. Wird beispielsweise eine Messreihe durchgeführt, zu deren Beginn die Last mit einer Spannung von 20 V zugeschaltet wird und anschließend in jedem Durchgang um 1 V erhöht wird, so dauert es etwa 2 Minuten, bis der Konverter nach dem Einschalten der Last den MPP gefunden hat, zwischen den einzelnen Messungen jedoch nur wenige Sekunden.

Dieser Tatsache wird in der Software Rechnung getragen, in dem – wie in Abschnitt 4.4.9 beschrieben – die Möglichkeit gegeben wird, sowohl eine Verzögerung vor der ersten Messung als auch zwischen den (folgenden) Messungen anzugeben.

Unter Anwendung der Überlegungen aus Abschnitt 3.5 (z. B. Gleichung 3.13) konnten einige Messunsicherheiten der Effizienzen – Konvertereffizienz, MPP-Tracking-Effizienz und Gesamteffizienz – bestimmt werden. Wie zu erwarten war, weist die Konvertereffizienz die größte Messunsicherheit auf, da sie von vier Messwerten abhängt (jeweils Strom und Spannung am Eingang wie am Ausgang), wogegen bei den anderen Effizienzen mit der mathematisch exakt definierten MPP-Leistung (gemäß Gleichung 3.1) gerechnet werden kann.

Für die Konvertereffizienz wurden in allen Tests Messunsicherheiten $e < \pm 0{,}001 = \pm 0{,}1\%$ berechnet (meist $e \approx \pm 0{,}0007 = \pm 0{,}07\%$). Verwendet man $e = \pm 0{,}001$, so ergibt sich beispielsweise für eine gemessene Konvertereffizienz von $\eta_{\text{mess}} = 95\%$ ein Intervall von $\eta = e \cdot \eta_{\text{mess}} = [94,905\% 95,095\%]$, in dem der tatsächliche Wert der Effizienz liegen wird.[23]

[23]Wertet man Gleichung 3.13 für die Extremwerte $e_{\text{mess},V_i} = 0{,}00005$, $e_{\text{Bereich},V_i} = 0{,}000035$, $V_{i,\text{Bereich,max}} = 0{,}1$ V, $V_i = 6$ mV, $I_{R_i} = 15$ A, $R_{i,\text{nom}} = 10\ \mu\Omega$, $R_{1,\text{th}} = 1$ K/W, $\overline{\alpha}_{R_i} = 0{,}00001$ K^{-1} und $e_{R_i,\text{Fertigung}} = 0{,}0002$ aus, so erhält man als obere Fehlergrenze $e_{\max} = 0{,}0024 \approx 0{,}24\%$. Da es sich bei den Annahmen jedoch um (physikalisch unmögliche) Extremwerte handelt, wird der Fehler immer signifikant kleiner sein als $e_{\max}$.

5.2 Testmessungen am Konverter SolarMagic SM3320-1A1

In erster Linie um die Implementation des Teststandes zu testen, aber auch, um etwas über den Konverter zu erfahren, wurden unter anderem folgende Testmessungen durchgeführt (Die Angaben von Strom und Spannung sind jeweils wie folgt zu lesen: V_{oc} / I_{sc} (V_{mpp} / I_{mpp})):

1. Einzelmessung 20 V / 2,5 A (16 V / 1,5 A), Lastspannung 25 V

2. Messreihe 20 V / 2,5 A (16 V / 1,5 A), Lastspannung 20 V $\leq V_L \leq$ 40 V, ΔV_L = 5 V

3. Messreihe 35 V / 7 A (25 V / 4 A), Lastspannung 12 V $\leq V_L \leq$ 32 V, ΔV_L = 1 V

4. Messreihe 35 V / 7 A (25 V / 6 A), Lastspannung 16 V $\leq V_L \leq$ 32 V, ΔV_L = 1 V

5. Messreihe 35 V / 8,5 A (25 V / 8 A), Lastspannung 24 V $\leq V_L \leq$ 35 V, ΔV_L = 1 V

Von diesen Messungen sollen die erste Messung als Beispiel für eine Einzelmessung sowie die fünfte Messung als Beispiel für eine Messreihe dargestellt werden. Es sei darauf hingewiesen, dass bei den Messungen zwei bis fünf jeweils nur *ein* Gleichspannungswandler im Messkreis angeschlossen war: Der Konverter C_1 wurde wie vorgesehen verkabelt, mit dem Unterschied, dass der negative Ausgang direkt auf die Last geführt wurde. Durch diesen „Kunstgriff“ ist es möglich, Konverter einzeln zu testen – auch solche, die *nicht* für den Einsatz an Solarmodulen konzipiert wurden.

Einzelmessung: MPP-Leistung 24 W

Wie bereits in Abschnitt 4.4.10 erwähnt, werden als Auswertung einer Einzelmessung sieben Graphen erstellt:

- Ein Graph, der den Verlauf aller Spannung über den Shunts darstellt (**Abbildung 5.1(a)**),

- je Konverter ein Graph, der die Verläufe von Eingangs- und Ausgangsspannung darstellt (**Abbildungen 5.1(b)** und **5.1(c)**),

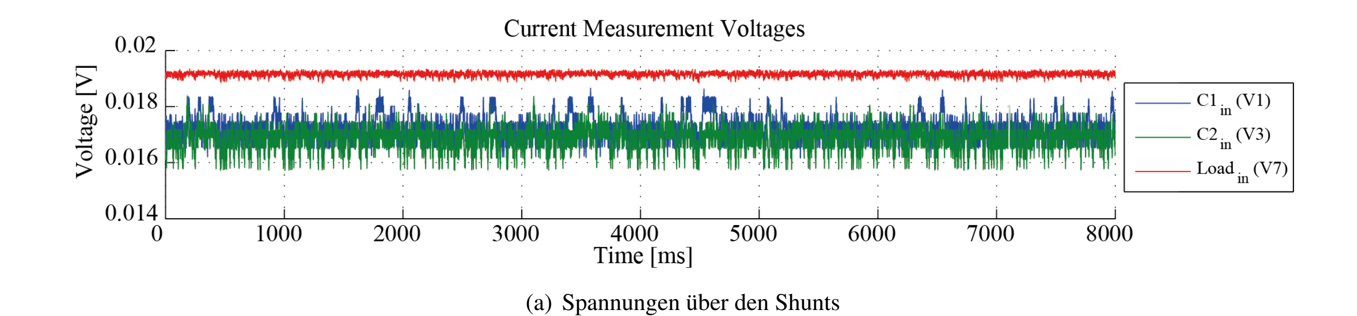
- je Konverter ein Graph, der den Konvertereingangsstrom gegenüber der Konvertereingangsspannung darstellt (die Solarkurve und der reale MPP werden ebenfalls angezeigt, **Abbildung 5.2**) sowie

- je Konverter ein Graph, der die Konvertereingangsleistung gegenüber der Konvertereingangsspannung darstellt (**Abbildung 5.3**).

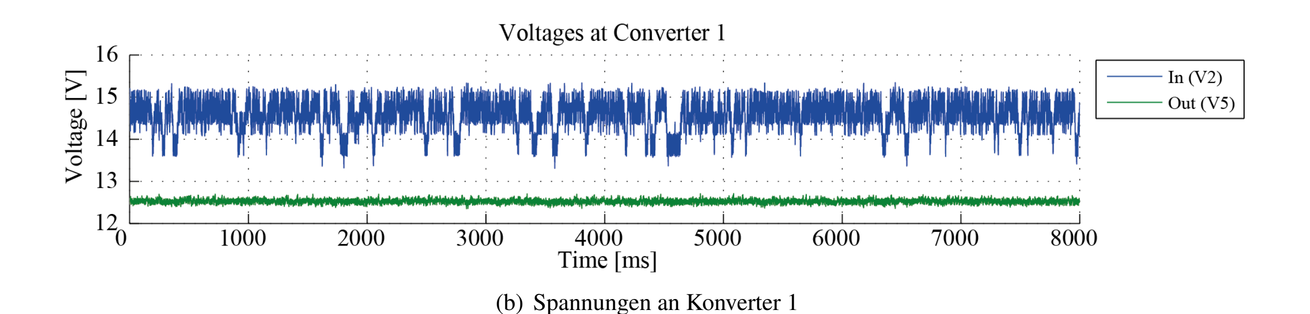
In den **Abbildungen 5.1(b)** und **5.1(c)** sind zwei Dinge deutlich zu erkennen: Einerseits zeigen die starken Schwankungen der Eingangsspannung beider Konverter die Arbeitsweise des MPP-Trackings[24], andererseits lässt sich aus der Tatsache, dass die Ausgangsspannung beider Konverter deutlich tiefer als ihre Eingangsspannung ist, schließen, dass beide Konverter im Tiefsetzstellerbetrieb arbeiten.

Abbildung 5.2 zeigt, dass die Konvertereingangsspannungen und -ströme immer auf der Solarkurve liegen und die Wertepaare, die effektiv angenommen werden, tatsächlich in der Nähe des (realen) MPPs liegen. Je geringer die Streuung dieser Messwerte ist, desto besser ist der MPP-Tracking-Algorithmus.

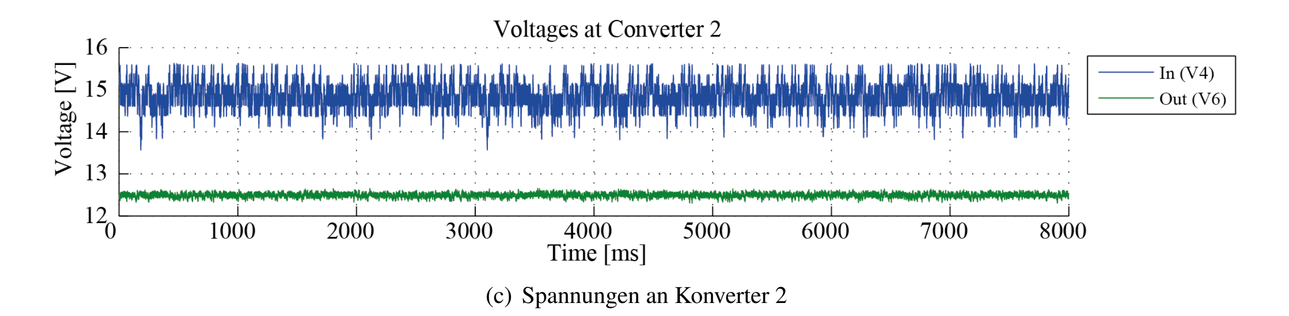
[24] Stufenweise Spannungsvariation, siehe Abschnitt 2.3



(a) Spannungen über den Shunts

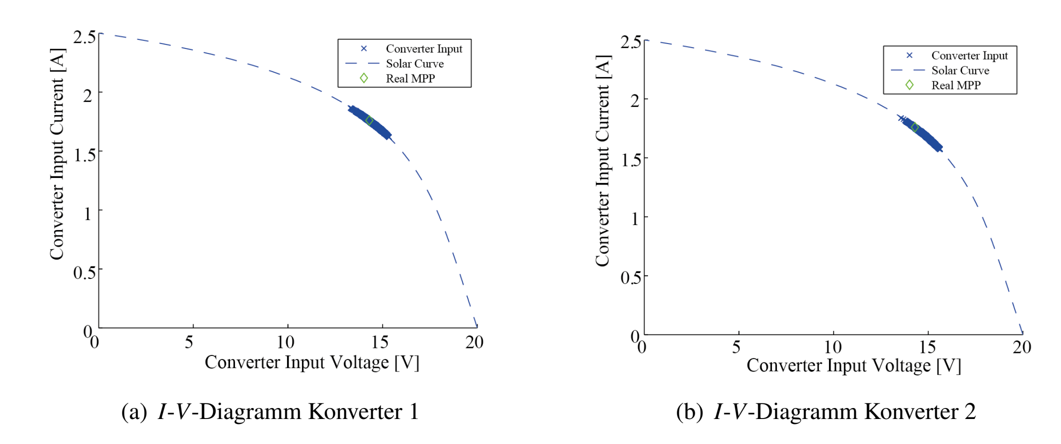


(b) Spannungen an Konverter 1



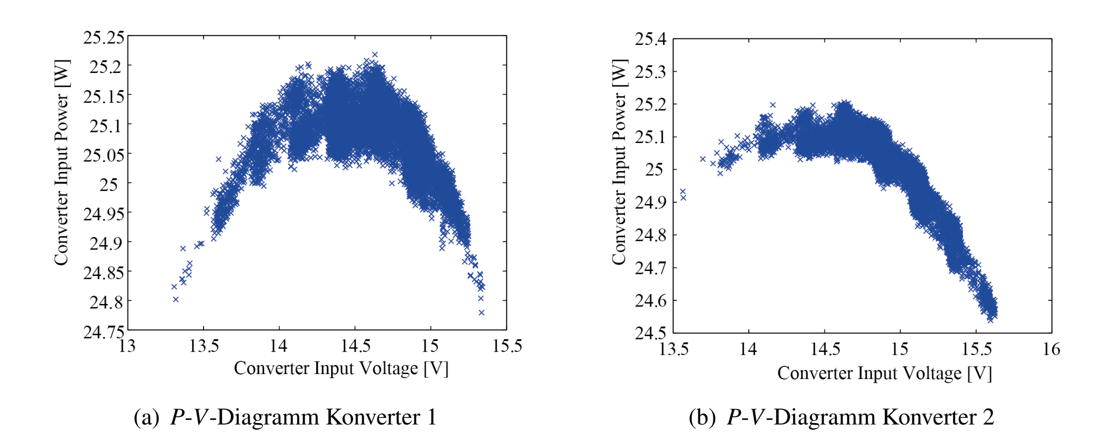
(c) Spannungen an Konverter 2

Abbildung 5.1: *Spannungsdiagramme der Auswertung einer Einzelmessung im ersten Fenster (figure1) für Testmessung 1*



(a) I-V-Diagramm Konverter 1 (b) I-V-Diagramm Konverter 2

Abbildung 5.2: *Strom-Spannungsdiagramme als Teil der Auswertung einer Einzelmessung im zweiten Fenster (figure2) für Testmessung 1*



(a) P-V-Diagramm Konverter 1 (b) P-V-Diagramm Konverter 2

Abbildung 5.3: *Leistung-Spannungsdiagramme als Teil der Auswertung einer Einzelmessung im zweiten Fenster (figure2) für Testmessung 1*

Abbildung 5.3 verdeutlicht nochmals die Bedeutung des MPPs: Je näher die Eingangsspannung an der MPP-Spannung liegt, desto höher ist auch die Eingangsleistung des Konverters[25].

Diese Graphen stellen alle gemessenen (respektive berechneten) Konvertereingangsleistungen gegenüber der Konvertereingangsspannung dar. Es lässt sich leicht abschätzen, wie viele Messpunkte nahe an der MPP-Spannung lagen; somit ist indirekt eine Aussage über die Qualität des MPP-Algorithmus (bei diesen Parametern) möglich.

Die Messwerte aller Spannungen sowie die berechneten Werte der Ströme und Effizienzen werden direkt in der Benutzeroberfläche angezeigt, **Abbildung 5.4** zeigt den *Arrangement Overview* mit den Ergebnissen dieser Testmessung.

[25] = aus dem Solarmodul gewonnene Leistung

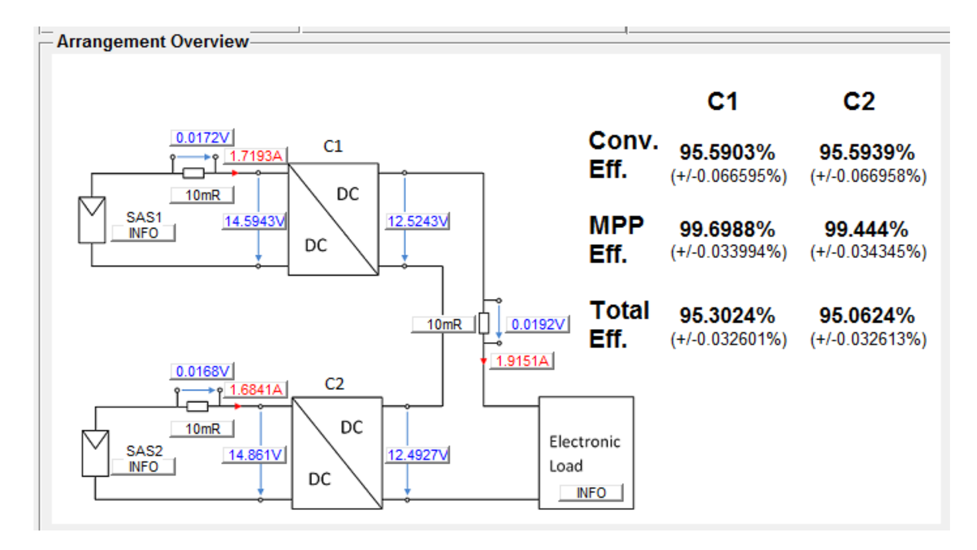


Abbildung 5.4: *Anzeige der Ergebnisse einer Einzelmessung (Serienschaltung) in der Benutzeroberfläche (Arrangement Overview) – für Testmessung 1*

Messreihe: MPP-Leistung 200 W

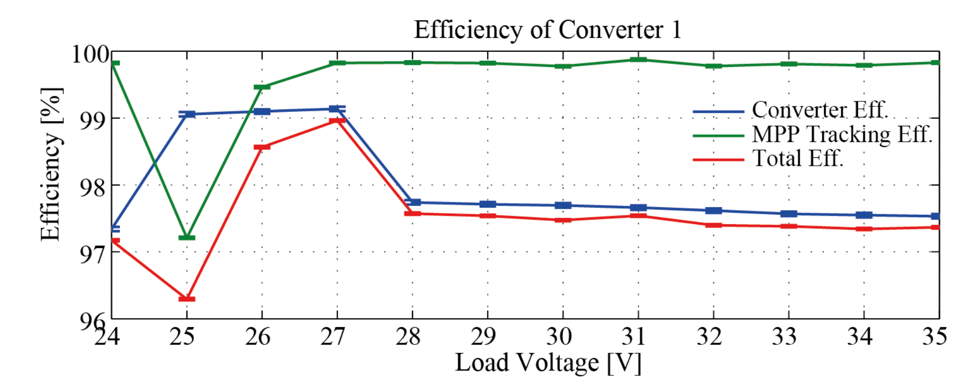
Auch die Auswertung von Messreihen wurde in Abschnitt 4.4.10 bereits kurz beschrieben, der Vollständigkeit halber sollen an dieser Stelle jedoch nochmals die Graphen aufgezählt werden, die nach Abschluss der Messreihe *automatisch* generiert werden:[26]

- Je Konverter ein Graph, in dem die drei Effizienzen Konvertereffizienz, MPP-Effizienz und Gesamteffizienz gegenüber der (eingestellten) Lastspannung darstellt werden (**Abbildung 5.5(a)**),
- je Konverter ein Graph, in dem die Konverterspannungen (Eingang und Ausgang) sowie die MPP-Spannung gegenüber der Lastspannung darstellt werden (**Abbildung 5.6(a)**),
- je Konverter ein Graph, in dem die Konverterströme (Eingang und Ausgang) sowie der MPP-Strom gegenüber der Lastspannung dargestellt werden (**Abbildung 5.6(b)**); in der zweiten Grafik
- je Konverter ein Graph, der die Konverterleistung (Eingang und Ausgang) sowie die MPP-Leistung gegenüber der Lastspannung darstellt (**Abbildung 5.7**) sowie
- je Konverter ein Graph, der die drei Effizienzen gegenüber der Konverterausgangsspannung darstellt (**Abbildung 5.5(b)**).

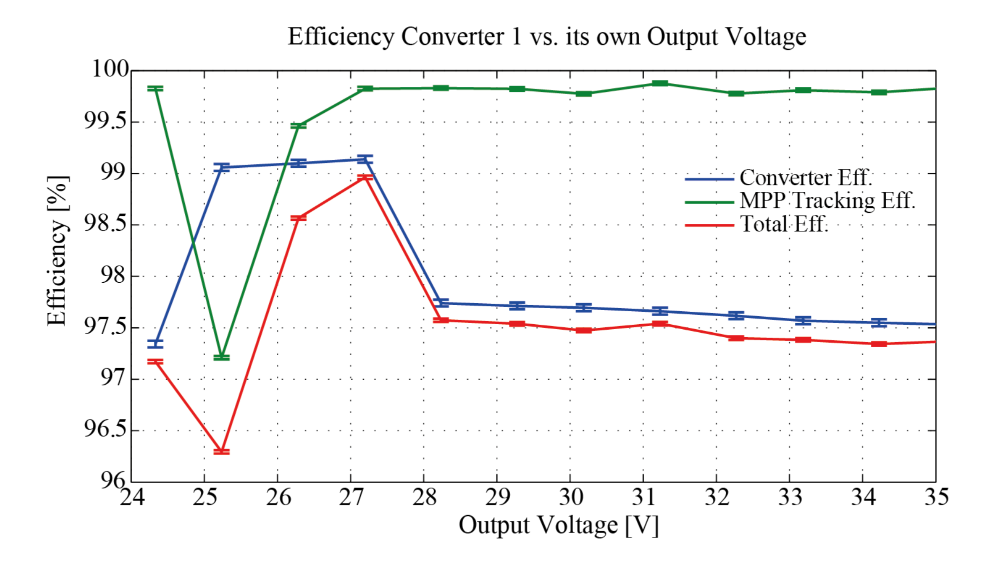
Insbesondere wenn nur ein Konverter getestet wird – wie in diesem Fall geschehen –, ermöglicht die Visualisierung der Effizienzen gegenüber der Lastspannung (**Abbildung 5.5(a)**) einen guten Eindruck bezüglich der Auslegung des Konverters. So wird in diesem Falle z. B. deutlich, dass der Konverter bei einer MPP-Leistung von (definiert) 200 W am effizientesten arbeitet, wenn die Ausgangsspannung zwischen 25 V und 27 V liegt – wobei bei einer Lastspannung von 25 V aus unerfindlichen Gründen der Durchschaltmodus aktiviert wird und die Effizienz der MPP-Trackings somit stark einbricht.

[26] Diese Funktionaliät wird erreicht, in dem die Funktion der Schaltfläche *Evaluate...* nach Abschluss der Messungen automatisch ausgeführt wird.

Soll eine ähnliche Aussage für einen Test mit mehreren Konvertern getätigt werden, so bietet sich die Darstellung der Effizienzen gegenüber der Konverterausgangsspannung (wie in **Abbildung 5.5(b)**) an. Wird nur ein Konverter getestet, so sind die Graphen Effizenzen-Lastspannung und Effizienzen-Konverterausgangsspannung nahezu identisch.[27]



(a) Konvertereffizienzen gegenüber der Lastspannung (*figure1*)



(b) Konvertereffizienzen gegenüber der Konverterausgangsspannung (*figure2*)

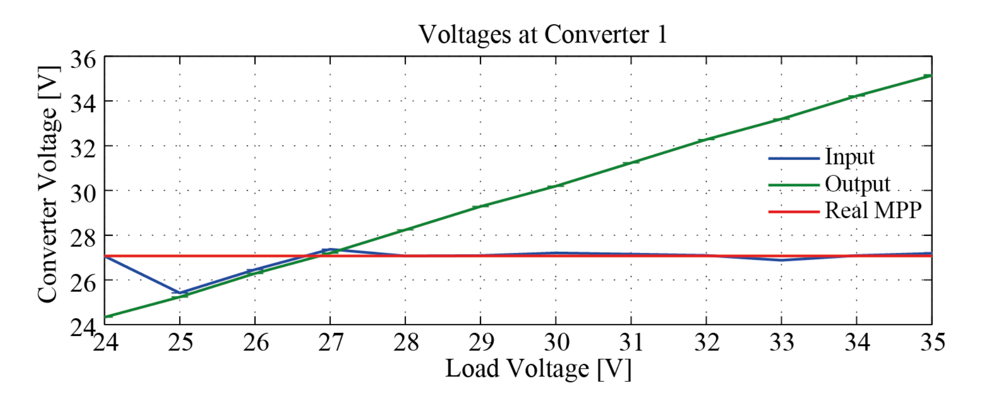
Abbildung 5.5: *Konvertereffizienzen gegenüber Lastspannung und Konverterausgangsspannung. Da in diesem Falle nur ein Konverter getestet wurde, sind die Graphen nahezu identisch.*[26]

Bei den Graphen Konverterspannungen-Lastspannung und Konverterströme-Lastspannung (**Abbildungen 5.6(a)** und **5.6(b)**) geht es wiederum um die Genauigkeit des MPP-Tracking-Algorithmus: Die Eingangsspannungen und -ströme der Konverter sollten für jede beliebige Lastspannung möglichst nahe am jeweiligen – ebenfalls eingetragenen – MPP-Wert liegen.

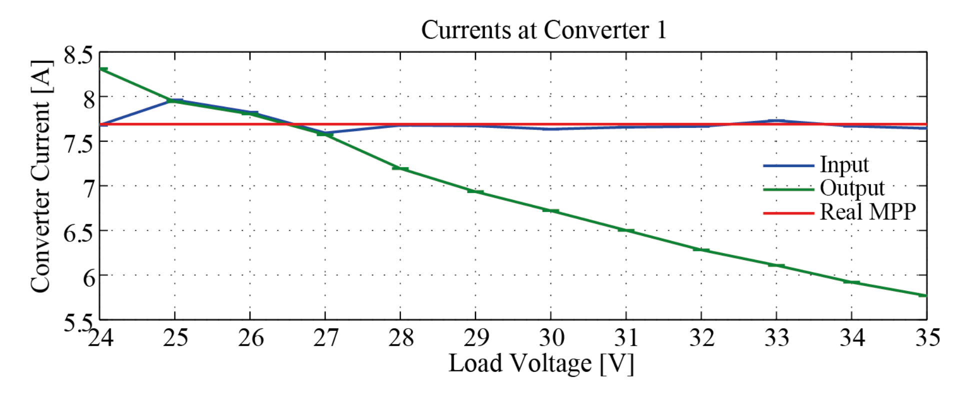
Des Weiteren ermöglichen diese Graphen eine Aussage über den Betriebsmodus des Konverters: Ist die Ausgangsspannung signifikant kleiner als die Eingangsspannung ($V_{\text{in}} < V_{\text{out}}$), so arbeitet der Konverter im Tiefsetzstellerbetrieb, ist die Ausgangsspannung signifikant größer als die Eingangsspannung ($V_{\text{in}} > V_{\text{out}}$), so arbeitet der Konverter im Hochsetzstellerbetrieb. Sind Eingangs- und Ausgangsspannung

[26] Der Spannungsabfall zwischen Konverterausgangsspannung und Lastspannung rührt vom Shunt her.

in etwa gleich groß ($V_{in} \approx V_{out}$)[27], so schaltet der Konverter bei dieser Lastspannung vom Eingang zum Ausgang durch (in **Abbildung 5.6(a)** ist dies bei 25 V, 26 V und 27 V Lastspannung der Fall).[28]



(a) Konverterspannungen gegenüber der Lastspannung



(b) Konverterströme gegenüber der Lastspannung

Abbildung 5.6: *Konverterspannungen und -ströme gegenüber der Lastspannung für eine Messreihe (Teil von figure1), in diesem Falle Testmessung 5*

Die Darstellung der Konverterleistung (Eingang und Ausgang sowie MPP-Leistung) gegenüber der Lastspannung verdeutlicht die Optimierung des Konverters auf einen Arbeitspunkt, an dem die Effizienz am größten ist – in diesem Graph an der Stelle, an dem die Ausgangsleistung am nächsten an der MPP-Leistung liegt. Auch die geringeren Verluste im Durchschaltmodus (in **Abbildung 5.7** bei 25 V, 26 V und 27 V) sind deutlich erkennbar.

[27] Man berücksichtige, dass vom Eingang zum Ausgang Verluste auftreten!

[28] Für den Strom gelten sinngemäß die gleichen Aussagen. Auf Grund der Relation $P = V \cdot I$ bedeutet $I_{in} < I_{out}$ aber einen *Tief*setzstellerbetrieb. Die anderen Betriebsarten ergeben sich gleichermaßen.

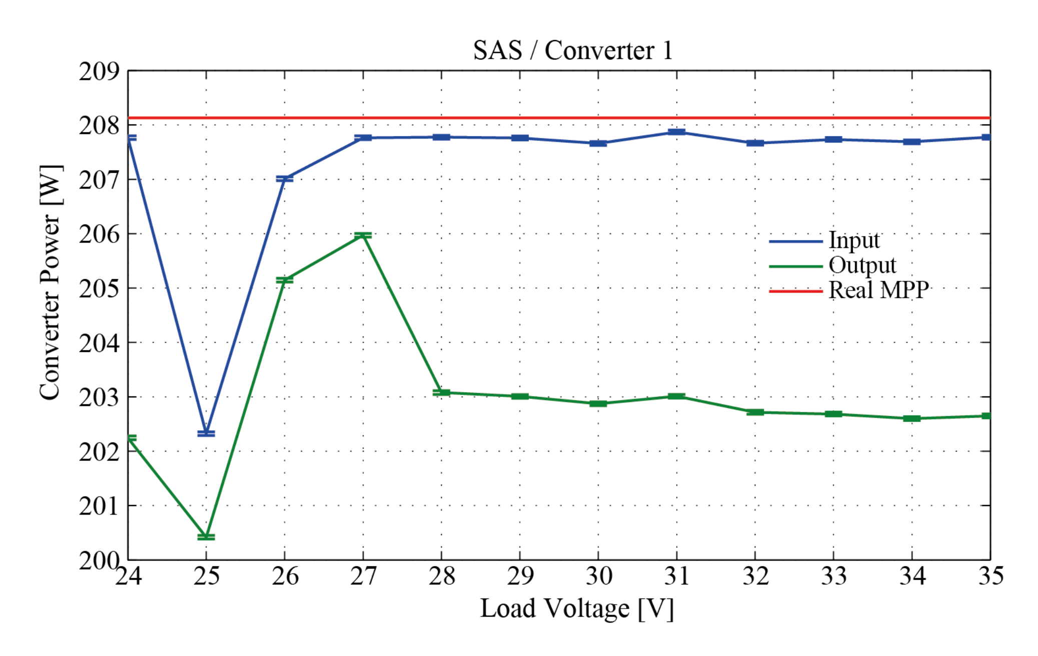


Abbildung 5.7: *Konverterleistungen gegenüber der Lastspannung bei einer Messreihe (Teil von figure2), hier für Testmessung 5*

6 Ausblick

Im Rahmen dieser Arbeit ist ein funktionsfähiger Teststand für solarmodulintegrierte Gleichspannungswandler entstanden, der auch für Tests von verschiedenen anderen Konvertertypen verwendet werden kann.[29] Des Weiteren wurden mit diesem Teststand einige Testmessungen an einem solarmodulintegrierten Konverter durchgeführt.

Die Ergebnisse dieser Arbeit – insbesondere der Teststand – lassen sich aus Sicht des Autors folgendermaßen weiterverwenden:

Es wäre wünschenswert, systematisch Messungen mit möglichst vielen der in den **Tabellen 2.1** und **2.2** aufgelisteten Konvertern durchzuführen. So kann einerseits der Teststand in seiner Funktionalität besser erprobt werden,[30] andererseits würde sich ein umfassende(re)s Bild über den aktuellen Stand der Technik ergeben. In einem späteren Schritt kann der Teststand dann dazu genutzt werden, einen eigenen Konverter unter verschiedensten Bedingungen zu testen und zu optimieren.

Bei der Software bestehen viele Erweiterungsmöglichkeiten. Eine aus Sicht des Autors besonders wertvolle Erweiterung der Software wäre durch den Ausbau der Messreihen zu erreichen: Wird nebst der Lastvariation – die selbstverständlich auf andere Modi wie CC, CR und CP erweitert werden könnte – eine SAS-Variation implementiert, so ergäbe sich die Möglichkeit eines viel umfassenderen Testens. Eine Auswertung der Messergebnisse in drei Dimensionen – z. B. Lastspannung auf der x-Achse, MPP-Spannung auf der y-Achse und Konvertereffizienz auf der z-Achse – würde für jeden Konvertertyp eine charakteristische Effizienz-Fläche ergeben und somit einen direkten Vergleich der verschiedenen Konverter ermöglichen.

Ebenso wäre es wünschenswert, gespeicherte Ergebnisse zwecks erneuter Auswertung laden zu können. Zwar besteht die Möglichkeit, die Messergebnisse in eine mat-Datei zu speichern, ein Laden und Wieder-Auswerten ist in der Benutzeroberfläche derzeit aber nicht möglich. Die Auswertung geladener Ergebnisse sollte sich nicht von der Auswertung einer gerade eben erfolgten Messung unterscheiden.

Weitere mögliche, kleine Erweiterungen oder / und Verbesserungen sollen nur stichwortartig aufgeführt werden:

- Zentralisierung der Zuordnung von Geräte-ID zu den Bezeichnungen der Messwerte
- Verbesserung der Fehlerbehandlung und Einrichten eines wirksamen Messabbruchs (ohne Neustart)
- Unterstützung der übrigen Modi der Last
- Verwendung der von MATLAB bestimmten Standardabweichung aller Spannungsmessungen zwecks Fehlerabschätzung
- Kompilieren des Programms in eine ausführbare Datei[31]

[29] Diese Konverter müssen nicht für den Betrieb im Zusammenhang mit Solaranlagen konzipiert sein.
[30] Auf Grund der wenigen und umständlichen Beschaffungswege für solarmodulintegrierte Konverter musste beispielsweise auf Tests mit der Parallelschaltung verzichtet werden.
[31] Es sollte die Verfügbarkeit der Treiber getestet werden!

Literaturverzeichnis

[Agilent SAS DB] AGILENT TECHNOLOGIES (Hrsg.): *Agilent E4360 Modular Solar Array Simulators : Models: E4360-62A, E4366-68A : Datasheet.* Santa Clara CA, 2011. – Datenblatt

[Agilent SAS HB] AGILENT TECHNOLOGIES (Hrsg.): *Modular Solar Array Simulator Series E4360 : User's Manual.* 5th Edition. Santa Clara CA, 2010. – Handbuch

[Agilent TMM DB] AGILENT TECHNOLOGIES (Hrsg.): *Agilent 34410A and 34411A Multimeters : Product Overview.* Santa Clara CA, 2007. – Datenblatt

[Agilent TMM HB] AGILENT TECHNOLOGIES (Hrsg.): *Agilent 34410A/11A Multimeter mit $6^1/_2$ Stellen : Benutzerhandbuch.* 3. Auflage. Santa Clara CA, 2006. – Handbuch

[BFE 2000] SCHWEIZERISCHE EIDGENOSSENSCHAFT – BUNDESAMT FÜR ENERGIE (Hrsg.): *Schweizerische Gesamtenergiestatistik 2000.* Bern, 2001. – Jahresstatistik

[BFE 2010] SCHWEIZERISCHE EIDGENOSSENSCHAFT – BUNDESAMT FÜR ENERGIE (Hrsg.): *Schweizerische Gesamtenergiestatistik 2010.* Bern, 2011. – Jahresstatistik

[Burster o.J.] BURSTER PRAEZISIONSMESSTECHNIK (Hrsg.): *Precision High Capacity Resistors : Model 1282 : Data Sheet.* Gernsbach, ohne Jahr. – Datenblatt

[Chroma HB] CHROMA ATE INC. (Hrsg.): *High Power DC Electronic Load 63200 Series : Operation & Programming Manual.* Kuei-Shan Hsiang, Taoyuan Hsien, 2009. – Handbuch

[DSolar o.J.] DEUTSCHESOLAR (Hrsg.): *Grundlagen der Photovoltaik.* Freiberg, ohne Jahr. – Firmenschrift

[Glotzbach 2009] GLOTZBACH, Thomas: *Grundlagen und Potenziale der Photovoltaik* (*14. Fuldaer Elektrotechnik-Kolloquium 2009*). Fulda, 2009. – Originalpräsentation

[Kranzer 2010] KRANZER, Dirk: *Module Integrated Electronics* (*OTTI-Seminar Power Electronics for Photovoltaics München 2010*). München, 2010. – Originalpräsentation

[MacLachlan 2011] MACLACHLAN, Derek ; KEITHLEY INSTRUMENTS (Hrsg.): *Getting Back to the Basics of Electrical Measurements.* Cleveland OH, 2011. – Firmenschrift

[VUWien 2012] INSTITUT FÜR MEDIZINISCHE PHYSIK (Hrsg.): *Messunsicherheit und Fehlerrechnung*. URL: http://www-med-physik.vu-wien.ac.at/physik/ws95/w9522dir/w9522120.htm. – Abgerufen am 08.03.2012 um 16:50 h. – Veterinäruniversität Wien

Datenblätter zu Tabelle 2.1

AZURAY TECHNOLOGIES, INC. (Hrsg.): *Azuray Power Optimizer Tech Sheet : Azuray AP250*. Durham OR, 2011. – Datenblatt

AZURAY TECHNOLOGIES, INC. (Hrsg.): *Azuray Power Optimizer Tech Sheet : Azuray AP260*. Portland OR, 2011. – Datenblatt

AZURAY TECHNOLOGIES, INC. (Hrsg.): *Azuray Power Optimizer Tech Sheet : Azuray AP300*. Portland OR, 2010. – Datenblatt

SOLAREDGE, LTD. (Hrsg.): *SolarEdge PowerBox : Module Add-On Box Solution : PB250-AOB, PB350-AOB*. Grass Valley CA, 2010. – Datenblatt

ST MICROELECTRONICS (Hrsg.): *STEVAL-ISV009V1 : 300 W photovoltaic converter demonstration board based on the SPV1020 : Data brief*. Ohne Ort, 2011. – Datenblatt

ST MICROELECTRONICS (Hrsg.): *SPV1020 : Interleaved DC-DC boost converter with built-in MPPT algorithm*. Ohne Ort, 2012. – Datenblatt

TIGO ENERGY, INC. (Hrsg.): *Tigo Energy Module Maximizer-ES (MM-ES) : Data Sheet*. Los Gatos CA, 2012. – Datenblatt

TEXAS INSTRUMENTS, INC. (Hrsg.): *SM1230, SM3320 : Application Note 2120 Power Optimizers Partial Deployment for Single String Systems*. Dallas TX, 2011. – Firmenschrift

Datenblätter zu Tabelle 2.2

EIQENERGY (Hrsg.): *vBOOST250 : 250-Watt DC-to-DC Converter Module*. San Jose CA, ohne Jahr. – Datenblatt

EIQENERGY (Hrsg.): *vBOOST350 : 350-Watt DC-to-DC Converter Module*. San Jose CA, ohne Jahr. – Datenblatt

TIGO ENERGY, INC. (Hrsg.): *Tigo Energy Module Maximizer-EP (MM-EP) : Data Sheet*. Los Gatos CA, 2011. – Datenblatt

Abbildungsverzeichnis

2.1 Schematisches Banddiagramm eines p-n-Übergangs 3
2.2 Beispielhafte I-V- und P-V-Kennlinien 4
2.3 Schaltung mit passiven Bypass-Dioden 4
2.4 Beispielhafte Solarkurve mit Kurve konstanter Leistung 5
2.5 Beispiele für Verschaltungen von Solarmodulen mit Gleichspannungswandlern 9

3.1 Schematischer Aufbau des Teststandes für Serien- und Parallelschaltung der Solarmodule 11
3.2 Schaltpläne des Teststandes für Serien- und Parallelschaltung der Gleichspannungswandler 13
3.3 Ausgang eines SAS-Moduls 15
3.4 Modellkurven des SAS 16

4.1 Hauptoberfläche des Teststandes vor dem Starten des Systems 25
4.2 SAS-Konfigurationsdialog 27
4.3 Last-Konfigurationsfenster 28
4.4 Fenster zur Messungskonfiguration (Trigger-Konfiguration) 29
4.5 Konfigurationsfenster für Messreihen 30
4.6 „Device Information“-Fenster für den Widerstand R_1 33
4.7 Benutzeroberfläche mit Anzeige des Ergebnisses einer Einzelmessung 33

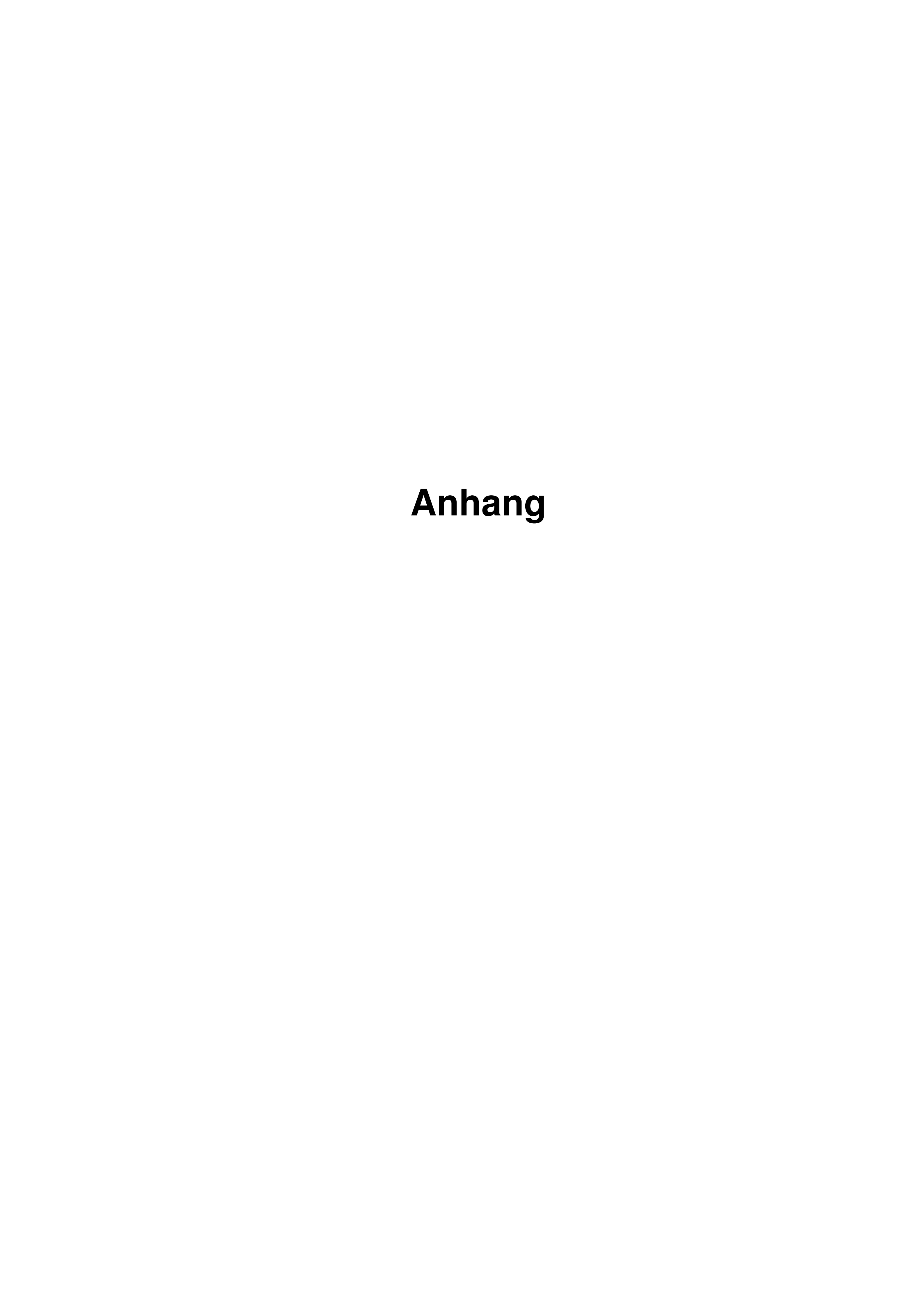
5.1 Spannungsdiagramme der Auswertung einer Einzelmessung 37
5.2 Strom-Spannungsdiagramme als Teil der Auswertung einer Einzelmessung 38
5.3 Leistung-Spannungsdiagramme als Teil der Auswertung einer Einzelmessung 38
5.4 Anzeige der Ergebnisse einer Einzelmessung in der Benutzeroberfläche 39
5.5 Konvertereffizienzen gegenüber Lastspannung und Konverterausgangsspannung 40
5.6 Konverterspannungen und -ströme gegenüber der Lastspannung für eine Messreihe . . . 41
5.7 Konverterleistungen gegenüber der Lastspannung bei einer Messreihe 42

Tabellenverzeichnis

2.1 Gleichspannungswandler für Serienschaltungen auf dem Markt 7
2.2 Gleichspannungswandler für Parallelschaltung auf dem Markt 8

3.1 Übersicht über die verwendeten Geräte und ihre Seriennummern 12

C.1 Technische Daten der SAS-Module des Typs E4361A-J01 bei einer Netz-Nennspannung von 230 V∼ 85
C.2 Daten der Agilent 34410A-Multimeter für Gleichspannungsmessungen 85
C.3 Auflösungen und Messgeschwindigkeiten in Abhängigkeit der NPLC für DC-Messungen 85
C.4 Unsicherheiten der Messungen für die Tischmultimeter 86
C.5 Daten der Hochpräzisionswiderstände 86
C.6 Wichtigste Daten der elektronischen Last 86



Anhang

A Software-Dokumentation

A.1 Überblick

Im Folgenden werden alle Funktionen – unterteilt in *Grafische Oberflächen*, kurz GUI[32], (Abschnitt A.2) und *Funktionen und Skripte* (Abschnitt A.4) – sowie alle globalen Variablen (Abschnitt A.3) der Steuerungssoftware des Teststands dokumentiert.

Es findet sich jeweils eine kurze Beschreibung der Aufgabe der Funktion oder der Variablen; außerdem eine Einteilung, um was für einen Variablen- oder Funktionstyp es sich handelt, zu welchem Programmbereich die jeweilige Funktion oder Variable gehört und wo die Variable definiert ist respektive wie die Funktion aufgerufen werden muss / kann.

Die folgende Auflistung umfasst die verwendeten Programmbereiche, die Bedeutung des betreffenden Programmbereichs wird jeweils kurz erläutert:

Einstellungen Hierzu gehören alle Funktionen, die *Grundeinstellungen* des Systems beinhalten und – falls es sich um systemweite Werte handelt – auch setzen. Diese Funktionen sind im Ordner *settings* zu finden.

Gültigkeit Diese Funktionen dienen dazu, Einstellungen, die in einer der GUIs vorgenommen werden können, auf ihre Gültigkeit zu überprüfen. Der zugehörige Ordner heißt *validity*.

Last Sämtliche Funktionen dieses Bereichs dienen dazu, mit der Last zu kommunizieren (Befehle senden, Werte lesen) und diese Werte in den entsprechenden Feldern der GUI `main` anzuzeigen. Die Funktionen befinden sich im Ordner *load*.

Multimeter Die Multimeter-Funktionen dienen einerseits der Kommunikation mit den Multimetern, andererseits aber auch der Auswertung der gemessenen Daten. Alle Funktionen dieses Bereichs liegen im Ordner *tmm*.

SAS Für diese Funktionen gilt sinngemäß das Gleiche wie für die Funktionen des Bereichs *Last*. Der zugehörige Ordner heißt *sas*.

System Zu diesem Bereich gehören alle Funktionen, die entweder das System als Ganzes betreffen (wie z. B. Starten oder Herunterfahren) oder von mehreren Funktionen des Systems genutzt werden. Alle diese Funktionen liegen im Ordner *system*.

Trigger Für die Funktionen des Bereichs *Trigger* gilt sinngemäß das Gleiche wie für die Funktionen des Bereichs *Last*; der zugehörige Ordner heißt *trig*.

Widerstand Die Funktion im Bereich *Widerstand* berechnet die Messunsicherheit auf Grund des Widerstandes. Sie befindet sich im Ordner *resistor*.

[32]Graphical User Interface, dt. Benutzeroberfläche

Bei den globalen Variablen des Typs struct findet sich meistens eine Tabelle, die die Felder der Variablen mit ihren Namen, Datentypen und Wertebereichen auflistet. Für Funktionen findet sich eine Tabelle oder eine Auflistung, die die Eingabeargumente genauer erklärt respektive verdeutlicht, welche Eingabeargumente zu welchen Ausgaben führen.

In den Funktionssignaturen ist jeweils der Name der Variablen angegeben, beispielsweise res = AskParam(elem, type), die Datentypen werden in der folgenden Aufzählung oder Tabelle der Argumente genannt. Folgende Datentypen werden verwendet:

ARR	–	Array	HDL	–	Handle	STC	–	Struct
BOOL	–	Boolean	INT	–	Integer	STR	–	String
CHR	–	Character	MIX	–	Verschiedene			
DBL	–	Double	OBJ	–	Objekt			

Man beachte, dass bei der Programmierung großer Wert auf die interne Dokumentation gelegt wurde. Aus diesem Grund ist zu jeder Funktion und jedem Skript (mit Ausnahme der GUI-Funktionen) eine Hilfe verfügbar, die im MATLAB-Befehlsfenster mittels help <Funktion>, für die Funktion AskParam also z. B. mittels help AskParam, aufgerufen werden kann.

A.2 Grafische Oberflächen

Allgemeiner Hinweis Es mag bei Betrachtung von Beschreibung und Quelltext der GUIs der Anschein entstehen, dass unnötige Funktionssignaturen vorhanden sind. Diese Funktionssignaturen *müssen* jedoch vorhanden sein, damit die GUI korrekt geladen werden kann – unabhängig davon, ob diese Funktionen etwas zur (implementierten) Funktionalität der GUI beitragen oder nicht!

ConfigMeasSeries

Die GUI ConfigMeasSeries dient dazu, eine Messreihe zu konfigurieren. Sie bietet die vorgesehenen Einstellungsmöglichkeiten und dient ebenso der Überprüfung der eingegeben Wert (siehe Abschnitt 4.4.9). Von den vorhandenen Callback-Funktionen wurden zwei für die Funktionalität der GUI explizit implementiert: CheckSettings_Callback führt nur einen Test der Einstellungen durch; StartSeries_Callback prüft zuerst die Gültigkeit der Eingaben, speichert (sofern alle Parameter gültig sind) die Einstellungen in der globalen Variablen series und schließt die GUI.[33]

Beide Callback-Funktionen verwenden zur Überprüfung die ebenfalls in ConfigMeasSeries.m implementierte Non-Callback-Funktion [ok,ans] = read(handles), die alle eingestellten Werte der GUI ausliest und auf ihre Gültigkeit hin überprüft. Sind die Werte gültig, dann nimmt ok den Wert 1 an, ans enthält die Werte, die in die Variable series geschrieben werden müssen – andernfalls hat ok den Wert 0, ans enthält einen String, der den Benutzer informiert, welche Werte nicht korrekt eingestellt sind.

[33] Während diese GUI geöffnet ist, befindet sich die Hauptroutine main.m im busy-wait-Zustand. Sobald die GUI ConfigMeasSeries geschlossen und die Variable series gesetzt ist, beginnt die Messreihe in main.m.

DevInfo

Diese GUI dient dazu, den Benutzer über die Eigenschaften des jeweiligen Gerätes beziehungsweise der jeweiligen physikalischen Größe zu informieren (siehe Abschnitt 4.4.13). Da es sich um eine reine Anzeige-GUI handelt, werden die Werte direkt beim Starten der GUI (`DevInfo_OutputFcn`) ausgelesen und in die handles der GUI geschrieben. Mittels der globalen Variablen `DevInf`, Feld `device`, wird der GUI mitgeteilt, für welches Gerät oder welche Größe die Informationen angezeigt werden sollen.

LoadConfig

In dieser GUI wird dem Benutzer die Möglichkeit geboten, die elektronische Last zu konfigurieren (siehe Abschnitt 4.4.7). Beim Starten der GUI (`LoadConfig_OutputFcn`) wird die direkt in der Datei `LoadConfig.m` implementierte Funktion `initLoadConfig(handles)` aufgerufen, die alle Einstellungen der Last liest und die Werte der zugehörigen Felder in der GUI entsprechend setzt.

Wird die Schaltfläche *Apply* betätigt, so wird die Funktion `Apply_Callback` ausgeführt. Diese prüft zuerst *für den ausgewählten Modus* (CC, CR, CV oder CP) die Gültigkeit der Einstellungen, bevor sie mittels der Funktionen `LoadSetMode` und `LoadChannel` die Werte an die Last sendet. Anschließend wird die GUI automatisch geschlossen.

main

Hierbei handelt es sich um die Hauptroutine des gesamten Teststandes. Durch den Aufruf der Funktion `Start` wird unter anderem auch diese GUI gestartet. Ihre Funktionen sind im Kapitel 4 beschrieben; eine vollständige Dokumentation würde ein ganzes Buch füllen.

Aus diesem Grunde seien hier nur einige Anmerkungen zur Funktionsweise gemacht: Einige der vorhandenen Callback-Funktionen, lange aber nicht alle, werden verwendet; z. B. jedes Mal dann, wenn eine Schaltfläche betätigt wird. Die Funktion `Refresh_Callback` dient dem Neuladen der GUI und wird deshalb von verschiedenen anderen Funktionen in der GUI aufgerufen.

Am Ende der Datei finden sich einige Non-Callback-Funktionen, die von verschiedenen Callback-Funktionen der GUI benötigt werden. Hierzu gehören die Vorbereitungsfunktion für die GUI `DevInfo`, die Funktion `ChangeArr` (um das Bild der Schaltung anzuzeigen und zu ändern) sowie die Funktion `HandleErr`(behandelt Fehler, welche einen Neustart des Systems erfordern).

SASconfig

Um den SAS wie in Abschnitt 4.4.5 zu konfigurieren, wird die GUI `SASconfig` benötigt. Im Gegensatz zur GUI `LoadConfig` werden hier alle Callback-Funktionen benutzt, die Werte jeweils in der globalen Variablen `values` abgelegt. In der Funktion `Apply_Callback` werden dann nur noch die Werte der Variablen `values` ausgelesen und auf ihre Gültigkeit überprüft, bevor entweder eine Warnung ausgegeben wird oder die Werte mittels der Funktionen `SASSetMode`, `SASSetFix` und `SASSetSAS` an den SAS gesendet werden.

Die Variable `values` wird nur in dieser GUI benötigt und deshalb beim Schließen vollständig gelöscht.

TrigConfig

Die GUI `TrigConfig`, deren Bedienung in Abschnitt 4.4.9 beschrieben ist, funktioniert genau analog zur GUI `LoadConfig` – mit dem einzigen Unterschied, dass zur Übermittlung der Werte die Funktionen `TMMInitAll` und `TrigSet` zum Einsatz kommen.

A.3 Globale Variablen

devices

Die Variable `devices` enthält Informationen über die verwendeten Geräte und ihre Konfiguration, namentlich Tischmultimeter, Funktionsgenerator, Solar Array Simulator und elektronische Last. Die Geräte werden dabei mittels Indizes angesprochen; so kann z. B. der Name des Gerätes mit dem Index (= programminterne Identifikationsnummer) 1 durch den folgenden Aufruf ermittelt werden:
`devices(1).name`

Typ: Struct
Klassifikation: Einstellungen
Definition: `VarDevices`

Feld	*Typ*	*Inhalt*	*conType*
name	STR	Instituts-Inventarbezeichnung	alle
serial	STR	Seriennummer des Geräts	alle
address	STR	VISA-Adresse des Geräts	USB
boardAddr	INT	GPIB-Board-Adresse	GPIB
primaryAddr	INT	Primäre GPIB-Adresse	GPIB
vendor	STR	Geräte-Hersteller	alle
conType	STR	Typ der Verbindung zwischen PC und Gerät	alle
visa	OBJ	Verbindungs-Objekt	alle
type	STR	Geräte-Typ (Bezeichnung)	alle

Anmerkung: Die Adresse, um ein Gerät via USB anzusprechen, setzt sich im Wesentlichen aus der Verkäufer-ID, der Geräte-ID und der Seriennummer des Gerätes zusammen. Für eine Verbindung mittels GPIB werden die GPIB-Board-Adresse sowie die Primäre GPIB-Adresse benötigt.

DevInf

Diese Variable dient dazu, der GUI `DevInfo` bei Aufruf aus `main` mitzuteilen, für welches Gerät oder welche physikalische Größe das Informationsfenster angezeigt werden soll.

Typ: Struct
Klassifikation: Wertübergabe an GUI
Defintion: `main.m`, Funktion `DeviceInfo`

ind

In der Variablen `ind` sind Informationen zu den bei der Variablen `devices` (s. o.) verwendeten Indizes gespeichert.
Wichtig: Funktionsvoraussetzung für das gesamte Programm ist, dass die Multimeter fortlaufend von `TMMmin` bis `TMMmax` (siehe weiter unten) nummeriert sind.

Typ: Struct
Klassifikation: Einstellungen
Defintion: `VarDevices`

Feld	*Typ*	*Inhalt*	*Wert*
TMMmin	INT	Kleinster für die Multimeter verwendeter Index	1
TMMmax	INT	Größter für die Multimeter verwendeter Index	7
TMMC1i	INT	Index des Multimeters am Eingang von Konverter 1	3
TMMC2i	INT	Index des Multimeters am Eingang von Konverter 2	6
TMMC1o	INT	Index des Multimeters am Ausgang von Konverter 1	2
TMMC2o	INT	Index des Multimeters am Ausgang von Konverter 2	5
TMMLiP	INT	Index des Multimeters am Last-Eingang in der Parallelschaltung	1
Trig	INT	Index des Triggers	8
SAS	INT	Index des Solar Array Simulators	9
Load	INT	Index der elektronischen Last	10
MAX	INT	Größter verwendeter Index = Anzahl Geräte	10

result

Die Variable `result` enthält die Ergebnisse der letzten Messung, *unabhängig* davon, ob es sich um eine Einzel- oder eine Serienmessung handelt. Die Variable wird jeweils mit dem Beginn einer neuen Messung gelöscht und neu erstellt.

Zwecks einfacherer Darstellung werden hier die Spannungen `v1` bis `v7` mit v_i abgekürzt, die Ströme `i1` und `i4` als i_i dargestellt[34] und die Konverter `c1` und `c2` unter c_i zusammengefasst. Alle Fehlerwerte sind grundsätzlich relative, einheitenfreie Werte. Um den Fehler in Prozent zu erhalten, muss also der Fehlerwert mit 100% multipliziert werden.

Typ: Struct
Klassifikation: Ergebnis
Defintion: `TMMReadAll`, `TMMEvalAll`

Tabelle mit den Feldern von `result` siehe nächste Seite.

[34] *Hinweis:* Im Falle einer Serienschaltung wird `i3` im Interface auch als I_L bezeichnet, `i4` existiert nicht.

Felder			*Typ*	*Inhalt*	*Wertebereich*
v_i	read[35]		ARR[DBL]	Alle gelesenen Werte der Spang. i	DBL $\in [0, 1000] \subset \mathbb{R}$ [V]
	val		DBL	Mittelwert aus allen Messwerten	$[0, 1000] \subset \mathbb{R}$
	dev[35]		DBL	Die von MATLAB berechnete Standardabweichung	$\mathbb{R}_{>0}$
	err		DBL	Messunsicherheit	$\mathbb{R}_{>0}$
vL[36]	val		ARR[DBL]	Werte der Lastspannung, die eingestellt wurden	DBL $\in [2{,}5, 500] \subset \mathbb{R}$
i_i	val		DBL	Mittelwert des Stromes i, berechnet aus zugehörigem Spannungs*mittelwert* und Widerstand	$\mathbb{R}$
	err		DBL	Messunsicherheit	$\mathbb{R}_{>0}$
c_i	ConvEff	val	DBL	SI-Wert der Effizienz vom Eingang zum Ausgang des Konverters i	$[0, 1] \subset \mathbb{R}$
		err	DBL	Messunsicherheit	$\mathbb{R}_{>0}$
	MPPEff	val	DBL	SI-Wert der Effizienz von der Quelle (SAS_i) zum Eingang von c_i	$[0, 1] \subset \mathbb{R}$
		err	DBL	Messunsicherheit	$\mathbb{R}_{>0}$
	TotEff	val	DBL	SI-Wert der Effizienz von der Quelle (SAS_i) zum Eingang der Last	$[0, 1] \subset \mathbb{R}$
		err	DBL	Messunsicherheit	$\mathbb{R}_{>0}$
int			DBL	Intervall zwischen den Messpunkten	$\mathbb{R}_{>0}$ [s]
type			STR	Typ der Messung	{`'single'`, `'L-Series'`}
arr			STR	Serien- oder Parallelschaltung der Konverter	{`'s'`,`'p'`}
delay[37]			DBL	Wartezeit zw. zwei Messungen	$\mathbb{R}_{>0}$ [s]
delay1[37]			DBL	Wartezeit vor der ersten Messung	$\mathbb{R}_{>0}$ [s]
SAS[38]	Voc		DBL	Leerlaufspannung des SAS-Kanals	$[0, 55] \subset \mathbb{R}$ [V]
	Isc		DBL	Kurzschlussstrom des SAS-Kanals	$[0, 8{,}5] \subset \mathbb{R}$ [A]
	Pmp		DBL	MPP-Leistung des SAS-Kanals[39]	$[0, 490] \subset \mathbb{R}$ [W]
	Vmp		DBL	MPP-Spannung des SAS-Kanals[39]	$[0, 50] \subset \mathbb{R}$ [V]
	Imp		DBL	MPP-Strom des SAS-Kanals[39]	$[0, 9] \subset \mathbb{R}$ [A]
SAS[40]	V		DBL	Spannung der Quelle	$[0, 50] \subset \mathbb{R}$ [V]
	I		DBL	Maximal zulässiger Strom	$[0, 9] \subset \mathbb{R}$ [A]

[36] Dieses Feld ist nur bei Einzelmessungen enthalten.
[37] Dieses Feld existiert nur, wenn es sich um eine Messreihe mit Lastspannungsvariation handelt.
[38] Dieses Feld existiert nur bei Messreihen.
[39] Zusammenfassung für die beiden Felder `SAS1` und `SAS2`. Diese Felder existieren nur, wenn der betreffende SAS-Kanal im SAS-Modus betrieben wurde.
[40] Es werden jeweils die realen (nicht die definierten) Werte abgelegt.
[41] Zusammenfassung für die beiden Felder `SAS1` und `SAS2`. Diese Felder existieren nur, wenn der betreffende SAS-Kanal im FIX-Modus betrieben wurde.

SASconf

Diese globale Variable wird beim Klicken der Schaltfläche *Configure* von *SAS 1 Control Panel* auf 1 gesetzt, bei Benutzung der gleichen Schaltfläche in *SAS 2 Control Panel* erhält sie den Wert 2. Sie dient dazu, der GUI `SASconfig` mitzuteilen, welcher Ausgang des SAS konfiguriert werden soll.

Typ: Struct
Klassifikation: Wertübergabe an GUI
Defintion: `main`

state

Die Variable `state` speichert den aktuellen Zustand des Systems (in erster Linie die Einstellungen der Geräte). Hinweis: Die Felder `TMM1` bis `TMM7` werden in der Tabelle der Einfachheit halber unter TMM_i, `SAS1` und `SAS2` unter SAS_i zusammengefasst.

Typ: Struct
Klassifikation: Zustandssicherung
Defintion: diverse

Felder		*Typ*	*Inhalt*	*Wertebereich*
arr		CHR	Verschaltung der Konverterausgänge	{'s','p'}
loadChannel		CHR	Aktiver Kanal der Last	{'A','B'}
nplc		DBL	Globale NPLC	{0,006, 0,02, 0,06, 0,2, 1, 2, 10, 100}
path		STR	Dateipfad, der zu `Start.m` führt	beliebig
SAS_i	mode	STR	Aktueller Modus des SAS_i	{'FIX','SAS'}
	Voc	DBL	Leerlaufspannung von SAS_i	$[0, 55] \subset \mathbb{R}$ [V]
	Isc	DBL	Kurzschlussstrom von SAS_i	$[0, 8{,}5] \subset \mathbb{R}$ [A]
	Vmp	DBL	MPP-Spannung von SAS_i	$[0, 50] \subset \mathbb{R}$ [V]
	Imp	DBL	MPP-Strom von SAS_i	$[0, 8{,}5] \subset \mathbb{R}$ [A]
	Pmp.def	DBL	Definierte MPP-Leistung von SAS_i	$[0, 490] \subset \mathbb{R}$ [W]
	Pmp.real	DBL	Reale MPP-Leistung von SAS_i[42]	? $(\subset \mathbb{R}_{>0})$
	Pmp.err	DBL	(nicht mehr verwendet)	
SystemTimeout		INT	Timeout-Zeit des Systems (nicht verwendet)	$\mathbb{R}_{>0}$ [s]
TMM_i	nplc[43]	DBL	NPLC für Multimeter *i*	{0,006, 0,02, 0,06, 0,2, 1, 2, 10, 100}
	range	DBL	Messbereich des Multimeters *i*	{0,1, 1, 10, 100, 1000} [V]
TMMtimer		DBL	Zeit zwischen zwei Samples der Multimeter	$\mathbb{R}_{\geq 0}$ [s]
trig	Dur	DBL	Dauer einer Messung	$\mathbb{R}_{>0}$ [s]
	n	INT	Anzahl Messungen pro Triggersignal	$[1, 50\,000]$
	pw	DBL	Pulsbreite des Triggersignals	$[10^{-6}, 2 \cdot 10^{-6}] \subset \mathbb{R}$ [s]
	v	DBL	Spannung des Triggersignals	$[3, 5] \subset \mathbb{R}$ [V]

[42] Vgl. hierzu Abschnitt 3.2.2
[43] Derzeit nicht benötigt, da alle Multimeter mit derselben NPLC arbeiten

values

Diese – nur in der GUI `SASconfig` verwendete – globale Variable dient einerseits dazu, die Konfiguration bei Aufruf der GUI wiederherstellen zu können, falls ein ungültiger Wert gesetzt werden soll – andererseits enthält sie die neu zu setzenden Werte, die überprüft werden sollen. Wird die GUI geschlossen, so wird auch die Variable gelöscht.

Typ: Struct
Klassifikation: Zustandssicherung
Defintion: `SASconfig`

A.4 Funktionen und Skripte

AskParam

Die Funktion `AskParam` dient dazu, den Wert eines bestimmten Parameters des Versuchsaufbaus zu ermitteln. Die Parameter können sich dabei auf Geräte (wie z. B. Widerstände), aber auch auf physikalische Größen (wie z. B. Ströme) beziehen.

Typ: Funktion
Klassifikation: Einstellungen
Signatur: `res = AskParam(elem, type)`

Ein- und Ausgabeargumente

Eingabeargument		*Ausgabeargument*			*Bemerkungen*
Var.	*Wertebereich*	*Var.*	*Typ*	*Wertebereich / Inhalt*	
elem	{'I1', 'I2', 'I3', 'I4', 'IL'}	res	INT	{-1}	Nicht verwendet
type	beliebig				
elem	{'Load'}	res	INT	Programminterne Identifikationsnummer der Last	
type	{'id'}				
elem	{'R1', 'R2', 'R3', 'R4'}	res	MIX	Eigenschaft des Widerstands	
type	{'alpha', 'err', 'exactVal', 'thermalR'}		DBL	Temperaturkoeffizient [44] Kalibrierungsfehler Genauer Widerstandswert Therm. Widerstand	[1/K] einheitenfrei [Ω] [K/W]
type	{'Calib', 'RegNo', 'val'}		STR	Kalibrationsnummer PES-Elementnummer Nominalwert	
elem	{'SAS1', 'SAS2'}	res	INT	Programminterne Identifikationsnummer des SAS	
type	{'id'}				
elem	{'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7'}	res	INT	Programminterne Identifikationsnummer des Multimeters	
type	{'id'}				

[44] In Gleichung 3.3 ist die Temperaturabhängigkeit der Widerstände gegeben. `alpha` entspricht α aus bezeichneter Gleichung; der Temperaturkoeffizient 2. Ordnung (β) wird vernachlässigt, da $\alpha \gg \beta$.

CheckFixMode

Diese Funktion überprüft, ob die angegeben Parameter für den Modus FIX des SAS gültig sind.

Typ: Funktion
Klassifikation: Gülitgkeit
Signatur: `[ok,mess] = CheckFixMode(VOLT,CURR)`

Ein- und Ausgabeargumente

`VOLT`	DBL	Eingestellte Spannung [V]
`CURR`	DBL	Eingestellte Strombegrenzung [A]
`ok`	BOOL	`true`, falls die Werte von `VOLT` und `CURR` zulässig sind
`mess`	STR	Text zur Anzeige in der GUI `SASconf` mit Hinweis auf den Fehler (falls `ok` == false)

CheckForError

Prüft, ob beim spezifizierten Gerät Error-Meldungen vorliegen, ruft diese ab und zeigt sie als Warnungen im Befehlsfenster an.

Typ: Funktion
Klassifikation: System
Signatur: `numOfErrors = CheckForError(devID,mfile)`

Ein- und Ausgabeargumente

`devID`	INT	Programminterne Identifikationsnummer des Geräts
`mfile`	STR	Name des Skripts, das diese Funktion aufgerufen hat
`numOfErrors`	INT	Anzahl ausgelesener Fehlermeldungen (0 ≤ `numOfErrors`)

CheckSASMode

Überprüft die Parameter für den SAS-Modus des SAS auf ihre Gültigkeit.

Typ: Funktion
Klassifikation: Gültigkeit
Signatur: `[ok,mess] = CheckSASMode(VOC,ISC,VMPP,IMPP)`

Ein- und Ausgabeargumente

`VOC`	DBL	Leerlaufspannung [V]
`ISC`	DBL	Kurzschlussstrom [A]
`VMPP`	DBL	MPP-Spannung [V]
`IMPP`	DBL	MPP-Strom [A]
`ok`	BOOL	`true`, falls alle Werte gültig sind
`mess`	STR	Text zur Anzeige in der GUI `SASconf` mit Hinweis auf den Fehler (falls `ok` == false)

CountAvailDev

Zählt die angeschlossenen und verbundenen Geräte.

Typ: Funktion
Klassifikation: System
Signatur: `num = CountAvailDev`

Ausgabeargument

`num`	INT	Anzahl verbundener Geräte

FixModeLimits

Gibt die Grenzwerte (obere / untere Grenze) der Einstellungen für den Modus FIX des SAS zurück.

Typ: Funktion
Klassifikation: Einstellungen
Signatur: `[u_max,u_min,i_max,i_min,p_max,p_min] = FixModeLimits`

Ausgabeargumente

`u_max`	DBL	Maximale zulässige Ausgangsspannung [V]
`u_min`	DBL	Minimale zulässige Ausgangsspannung [V]
`i_max`	DBL	Maximaler zulässiger Ausgangsstrom [A]
`i_min`	DBL	Minimaler zulässiger Ausgangsstrom [A]
`p_max`	DBL	Maximale zulässige Ausgangsleistung [W]
`p_min`	DBL	Minimale zulässige Ausgangsleistung [W]

IsConnected

Prüft, ob eine Verbindung zu einem bestimmen Gerät besteht.

Typ: Funktion
Klassifikation: System
Signatur: `status = IsConnected(devID)`

Ein- und Ausgabeargumente

`devID`	INT	Programminterne Identifikationsnummer des Geräts
`status`	BOOL	`true`, falls das Gerät verbunden ist, sonst `false`

LoadChannel

Aktiviert einen der beiden Kanäle der Last.

Typ: Funktion
Klassifikation: Last
Signatur: `success = LoadChannel(channel)`

Ein- und Ausgabeargumente

`channel`	CHR	Der gewünschte Kanal (`'A'` oder `'B'`)
`success`	BOOL	`true`, falls der Kanal fehlerfrei gewählt werden konnte, sonst `false`

LoadCheckParams

Prüft die gewünschten Einstellungen der Last auf ihre Gültigkeit.

Typ: Funktion
Klassifikation: Gültigkeit
Signatur: `[ok,mess] = LoadCheckParams(mode,A,B,add1,add2)`

Ein- und Ausgabeargumente

`mode`	STR	Betriebsmodus der Last (`'CC'`, `'CR'`, `'CV'` oder `'CP'`)
`A`	DBL	Einzustellender Wert für Kanal A (in SI-Basiseinheit)
`B`	DBL	Einzustellender Wert für Kanal B (in SI-Basiseinheit)
`add1`	MIX	Anstiegsrate oder Maximalstrom (CV) als DBL, STR 'max' für CV möglich
`add2`	MIX	Abfallrate als DBL für CC, CR und CP; BOOL für CV (Fast Mode on: true/false)
`ok`	BOOL	`true`, falls alle Werte gültig sind, sonst `false`
`mess`	STR	Text zur Anzeige in GUI `LoadConfig` mit Hinweis auf Fehler (falls `ok` == false)

LoadDefaults

Gibt die Standardeinstellungen für die Last zurück.

Typ: Funktion
Klassifikation: Einstellungen
Signatur: `[mode,range,A,B,add1,add2,ch] = LoadDefaults`

Ausgabeargumente

`mode`	STR	Standard-Betriebsmodus (CC, CR, CV oder CP)
`range`	STR	Standardbereich (HIGH oder LOW)
`A`	DBL	Standardwert für Kanal A (in SI-Basiseinheit)
`B`	DBL	Standardwert für Kanal B (in SI-Basiseinheit)
`add1`	MIX	Anstiegsrate (CC, CR, CP) als DBL oder Strombegrenzung als DBL / STR 'max' (CV)
`add2`	MIX	Abfallrate (CC, CR, CP) als DBL oder „Fast Mode on“ (CV) als BOOL
`ch`	CHR	Standardmäßig aktiver Kanal (A oder B)

LoadLimits

Gibt die Maximal- und Minimalwerte der Last (obere / untere Grenze) zurück.

Typ: Funktion
Klassifikation: Einstellungen
Signatur: `[Min,Max,RISEmin,RISEmax,FALLmin,FALLmax] = LoadLimits(mode)`
`[Vmin,Vmax,IMAXmin,IMAXmax] = LoadLimits('CV')`

Ein- und Ausgabeargumente

`mode`	STR	Modus, für den die Grenzwerte abgefragt werden sollen (CC, CR oder CP)
`Min`	DBL	Zulässiger Minimalwert in SI-Basiseinheit
`Max`	DBL	Zulässiger Maximalwert in SI-Basiseinheit

`RISEmin`	DBL	Minimale Anstiegsrate; [A/μs] (CC, CP) oder [A/μS] (CR)
`RISEmax`	DBL	Maximale Anstiegsrate, Einheiten wie oben
`FALLmin`	DBL	Minimale Abfallrate, Einheiten wie oben
`FALLmax`	DBL	Maximale Abfallrate, Einheiten wie oben
`Vmin`	DBL	Minimale Spannung [V]
`Vmax`	DBL	Maximale Spannung [V]
`IMAXmin`	DBL	Minimale Strombegrenzung [A]
`IMAXmax`	DBL	Maximale Strombegrenzung [A]

LoadQuery

Sendet eine Anfrage an die Last und gibt das Ergebnis der Anfrage zurück.

Typ: Funktion
Klassifikation: Last
Signatur: `[...] = LoadQuery(query)`

Ein- und Ausgabeargumente

query	*Ausgabeargumente*		
	Var.	*Typ*	*Inhalt*
`'OUT'`	on	BOOL	`true`, falls die Last eingeschaltet ist, sonst `false`
`'MODE'`	mode	STR	Aktueller Betriebsmodus (CC, CR, CV oder CP)
	range	STR	Eingestellter Bereich (HIGH oder LOW)
`'PAR'`	valA	DBL	Eingestellter Wert für Kanal A [`unit`]
	valB	DBL	Eingestellter Wert für Kanal B [`unit`]
	unit	CHR	Einheit der Werte `valA` und `valB` (A, R, V, W)
	add	MIX	Strombegrenzung für CV als DBL, sonst STR 'err'
`'CURR'` `'RES'` `'POW'`	i1 / r1 / p1	DBL	Eingestellter Wert für Kanal A im entsprechenden Modus (CC / CR / CP) [A] / [Ω] / [W]
	i2 / r2 / p2	DBL	Eingestellter Wert für Kanal B im entsprechenden Modus (CC / CR / CP) [A] / [Ω] / [W]
	rise	DBL	Eingestellte Anstiegsrate für den entsprechenden Modus (CC / CR / CP) [A/μs] / [A/μS] / [A/μs]
	fall	DBL	Eingestellte Abfallrate für den entsprechenden Modus (CC / CR / CP) [A/μs] / [A/μS] / [A/μs]
`'VOLT'`	v1	DBL	Eingestellte Spannung für Kanal A im CV-Modus [V]
	v2	DBL	Eingestellte Spannung für Kanal B im CV-Modus [V]
	imax	DBL	Eingestellte Strombegrenzung für den CV-Modus [A]
	fast	BOOL	Fast Mode aktiv oder nicht
`'MEAS'`	i	DBL	Aktueller Strommesswert [A]
	p	DBL	Aktuell gemessene Leistung [W]
	r	DBL	Aktuell gemessenener Widerstand [Ω]
	v	DBL	Aktueller Spannungsmesswert [V]

LoadSetDefaults

Diese Funktion setzt die Einstellungen der Last auf die Standardwerte.

Typ: Funktion
Klassifikation: Last
Signatur: `success = LoadSetDefaults`

und Ausgabeargument

`success`	BOOL	`true`, falls die Standardeinstellungen gesetzt werden konnten

LoadSetMode

Aktiviert den ausgewählten Modus für die Last.

Typ: Funktion
Klassifikation: Last
Signatur: `success = LoadSetMode(mode,range)`

Ein- und Ausgabeargumente

`mode`	STR	Gewünschter Modus ('CC', 'CR', 'CV' oder 'CP')
`range`	STR	'HIGH' oder 'LOW'
`success`	BOOL	`true`, falls der gewünschte Modus gesetzt wurde

LoadSetParams

Setzt für einen Betriebsmodus die Parameter der Last.

Typ: Funktion
Klassifikation: Last
Signatur: `success = LoadSetParams(mode,A,B<,add1,add2>)`

Ein- und Ausgabeargumente

<table>
<tr><th colspan="5">Eingabeargument</th><th rowspan="2">Bedeutung</th></tr>
<tr><th>mode</th><th>A</th><th>B</th><th>add1</th><th>add2</th></tr>
<tr><td rowspan="2">CC / CR / CP</td><td rowspan="2">DBL</td><td rowspan="2">DBL</td><td></td><td></td><td>Setzt die Werte A und B für den gewählten Modus [A] / [Ω] / [W]</td></tr>
<tr><td>DBL</td><td>DBL</td><td>Setzt zusätzlich die Anstiegs- (add1) und Abfallrate (add2) [A/μs] / [A/μS] / [A/μs]</td></tr>
<tr><td rowspan="2">CV</td><td rowspan="2">DBL</td><td rowspan="2">DBL</td><td></td><td></td><td>Setzt die Spannungen A und B für die betreffenden Kanäle [V]</td></tr>
<tr><td>DBL</td><td>BOOL</td><td>Setzt die Strombegrenzung auf add1 [A] und „Fast Mode on“ auf add2</td></tr>
</table>

LoadSetStatus

Diese Funktion setzt die handles der GUI `main`, die über den Zustand der Last Aufschluss geben.

Typ: Funktion
Klassifikation: Last
Signatur: `LoadSetStatus(handles)`

Eingabeargument

`handles`	HDL	handles aus der GUI `main`

LoadSwitch

Schaltet die Last an oder aus.

Typ: Funktion
Klassifikation: Last
Signatur: `success = LoadSwitch(action)`

Ein- und Ausgabeargumente

`action`	STR	'ON' zum Einschalten, 'OFF' zum Ausschalten
`success`	BOOL	`true`, falls die Schaltaktion durchgeführt werden konnte

MakeDate

Erstellt einen String des aktuellen Datums.

Typ: Funktion
Klassifikation: System
Signatur: `date = MakeDate(format)`

Ein- und Ausgabeargumente

`format`	STR	Gewünschtes Format (yyyymmdd, yymmdd, yyyy-mm-dd, yy-mm-dd, dd.mm.yyyy, dd.mm.yy)
`date`	STR	Aktuelles Datum im gewünschten Format

MakeTimestamp

Erstellt einen aktuellen Zeitstempel.

Typ: Funktion
Klassifikation: System
Signatur: `timestamp = MakeTimestamp`

Ausgabeargument

`timestamp`	STR	Zeitstempel im Format 'dd.mm.yyyy hh:mm:ss'

Report

Schreibt eine Meldung in das Befehlsfenster und das tägliche Logfile.

Typ: Funktion
Klassifikation: System
Signatur: `Report(text,type)`

Eingabeargumente

`text`	STR	Zu verarbeitende Meldung
`type`	CHR	Typ der Meldung ('E', 'W', 'I')

ReportError

Schreibt einen ERROR ins Befehlsfenster und das Logfile und wirft eine MException.

Typ: Funktion
Klassifikation: System
Signatur: `ReportError(text)`

Eingabeargument

`text`	STR	Die Meldung

ReportInfo

Schreibt eine Information ins Befehlsfenster und das Logfile.

Typ: Funktion
Klassifikation: System
Signatur: `ReportInfo(text)`

Eingabeargument

`text`	STR	Die Meldung

ReportWarning

Schreibt eine Warnung ins Befehlsfenster und das Logfile.

Typ: Funktion
Klassifikation: System
Signatur: `ReportWarning(text)`

Eingabeargument

`text`	STR	Die Meldung

ResError

Berechnet die Messungenauigkeit eines Widerstands.

Typ: Funktion
Klassifikation: Widerstand
Signatur: `error = ResError(name,curr)`

Ein- und Ausgabeargumente

`name`	STR	Bezeichnung des Widerstands ('R1', 'R2', 'R3' oder 'R4')
`curr`	DBL	Gemessener Strom durch den Widerstand [A]
`error`	DBL	Relative Messunsicherheit (einheitenfrei)[45]

ResetMeasureHandlesValues

Setzt alle Anzeigen des *Arrangement Overview* zurück auf die symbolischen Bezeichnungen.

Typ: Funktion
Klassifikation: System
Signatur: `ResetMeasureHandlesValues(handles)`

Eingabeargument

`handles`	HDL	handles der GUI `main`

RoundValue

Rundet einen gegebenen Wert.

Typ: Funktion
Klassifikation: System
Signatur: `rounded = RoundValue(value,accuracy)`

Ein- und Ausgabeargumente

`value`	DBL	Exakter, zu rundender Wert
`accuracy`	STR	'2PERCENT' rundet den Wert gemäß Vorgabe[46] und rechnet in Prozent um,
	STR	'NORM' rundet den Wert gemäß Vorgabe,
	STR	'LIVE' rundet den Wert auf zwei Stellen weniger als vorgegeben,
	INT	rundet den Wert auf `accuracy` Nachkommastellen
`rounded`	DBL	Der gerundete Wert

[45] Der Prozentwert ist duch Multiplikation mit 100% zu bestimmen.

SASDefaults

Gibt die Standardeinstellungen für den SAS zurück.

Typ: Funktion
Klassifikation: Einstellungen
Signatur: `[mode,volt,curr,voc,isc,vmpp,impp] = SASDefaults`

Ausgabeargumente

`mode`	STR	Standard-Betriebsmodus des SAS ('FIX' oder 'SAS')
`volt`	DBL	FIX-Spannung [V]
`curr`	DBL	FIX-Strombegrenzung [A]
`voc`	DBL	Leerlaufspannung im SAS-Modus [V]
`isc`	DBL	Kurzschlussstrom im SAS-Modus [A]
`vmpp`	DBL	MPP-Spannung im SAS-Modus [V]
`impp`	DBL	MPP-Strom im SAS-Modus [A]

SASDetermineModel

Berechnet die simulierte Solarkurve nach Gleichung 3.1 und stellt sie entweder grafisch dar oder gibt gefragte Werte der Kurve zurück.

Typ: Funktion
Klassifikation: SAS
Signatur: `SASDetermineModel`
`[...] = SASDetermineModel(property,channel)`

Ein- und Ausgabeargumente

Eingabeargumente		*Ausgabe(argumente)*	
STR property	*INT channel*	*Typ*	*Inhalt*
		–	Keine Rückgabewerte; Erstellung einer Grafik mit je einem Plot für SAS_1 und SAS_2
'RealMPP'	{1, 2}	DBL	Real simulierte MPP-Spannung [V]
		DBL	Real simulierter MPP-Strom [A]
'RealPmp'	{1, 2}	DBL	Real simulierte MPP-Leistung [W]
'SolarCurve'	{1, 2}	ARR[DBL]	Alle Spannungen, die auf der Solarkurve liegen und mittels Gleichung 3.1 berechnet wurden
		ARR[DBL]	Alle Ströme `I=0:.01:I`$_{sc}$, die zur Berechnung der Solarkurve verwendet wurden

SASModeLimits

SASModeLimits gibt die Begrenzungen des SAS-Modus zurück.

Typ: Funktion
Klassifikation: SAS
Signatur: `[...] = SASModeLimits`

Ausgabeargumente (in der korrekten Reihenfolge,[47] alle vom Typ DBL)

- Maximale Leerlaufspannung [V]
- Minimale Leerlaufspannung [V]
- Maximaler Kurzschlussstrom [A]
- Minimaler Kurzschlussstrom [A]
- Maximale MPP-Spannung [V]
- Minimale MPP-Spannung [V]
- Maximaler MPP-Strom [A]
- Minimaler MPP-Strom [A]
- Maximale Ausgangsleistung [W]
- Minimale Ausgangsleistung [W]
- Minimaler Widerstand (dV/dI) für $V > V_{mpp}$

SASQuery

Sendet eine Anfrage an den SAS und gibt die Antwort zurück.

Typ: Funktion
Klassifikation: SAS
Signatur: `[ch1,ch2] = SASQuery(query)` // `set = SASQuery(query,channel)`

Der erste Aufruf führt die Abfrage für beide Kanäle durch, der zweite Aufruf nur für Kanal `channel`.

Ein- und Ausgabeargumente

`query`	STR	Anfrage, nicht case-sensitive (siehe Tabelle weiter unten)
`channel`	INT	Anfrage auf diesen Kanal beschränken (1 oder 2)
`ch1`	MIX	Die Antwort von Kanal 1
`ch2`	MIX	Die Antwort von Kanal 2
`set`	MIX	Die Antwort des spezifizierten Kanals

query	*Ausgabeargument(e)*	
	Typ	*Inhalt*
OUT	BOOL	Zustand des Ausgangs (1 = ON)
MODE	STR	Modus des Ausgangs ('FIX', 'SAS', 'TABL', 'unknown')
STATUS	STR	Einstellungen (Spannung/Strom) des Ausgangs
Voc	DBL	Leerlaufspannung
Isc	DBL	Kurzschlussstrom
Vmpp	DBL	Definierte MPP-Spannung
Impp	DBL	Definierter MPP-Strom

[47] Zuerst die linke Spalte von oben nach unten, dann die rechte ebenso

SASReadMeasurement

Liest die aktuellen Strom- und Spannungswerte am SAS-Ausgang und gibt sie zurück.

Typ:	Funktion
Klassifikation:	SAS
Signatur:	`[v1,i1,v2,i2] = SASReadMeasurement` `[v,i] = SASReadMeasurement(channel)`

Der erste Aufruf liest beide Kanäle aus, der zweite nur Kanal `channel`.

Ein- und Ausgabeargumente

`channel`	INT	Auszulesender Kanal (1 oder 2, sofern nicht beide gelesen werden sollen)
`v1`	DBL	Spannung am Ausgang 1 [V]
`i1`	DBL	Strom am Ausgang 1 [A]
`v2`	DBL	Spannung am Ausgang 2 [V]
`i2`	DBL	Strom aus Ausgang [A]
`v`	DBL	Spannung am spezifizierten Ausgang [V]
`i`	DBL	Strom am spezifizierten Ausgang [A]

SASReadParams

Liest die derzeitigen Einstellungen des SAS und gibt sie zurück.

Typ:	Funktion
Klassifikation:	SAS
Signatur:	`[mode,v,i,voc,isc,vmpp,impp] = SASReadParams(<channel>)`

Ein- und Ausgabeargumente

`channel`	INT	Auswahl des Kanals (1 oder 2)
`mode`	ARR[STR] / STR	Modi beider / des ausgewählten Kanals
`v`	ARR[DBL] / DBL	FIX-Spannungen beider / des ausgewählten Kanals [V]
`i`	ARR[DBL] / DBL	FIX-Strombegrenzung [A]
`voc`	ARR[DBL] / DBL	Leerlaufspannung [V]
`isc`	ARR[DBL] / DBL	Kurzschlussstrom [A]
`vmpp`	ARR[DBL] / DBL	MPP-Spannung [V]
`impp`	ARR[DBL] / DBL	MPP-Strom [A]

SASSetDefaults

Sendet die Standardeinstellungen an den SAS.

Typ: Funktion
Klassifikation: SAS
Signatur: success = SASSetDefaults(<channel>)

Ist channel angegeben, so wird nur der betreffende Kanal zurückgesetzt, ansonsten beide.

Ein- und Ausgabeargumente

channel	INT	Zurückzusetzender Kanal (1 oder 2)
success	BOOL	true, falls die Operation durchgeführt wurde

SASSetFix

Sendet die Einstellungen für den FIX-Modus an den SAS *ohne* den FIX-Modus zu aktivieren.

Typ: Funktion
Klassifikation: SAS
Signatur: success = SASSetFix(volt,curr<,channel>)

Ist channel angegeben, werden die Einstellungen nur an den entsprechenden Kanal gesendet, ansonsten an beide.

Ein- und Ausgabeargumente

volt	DBL	Einzustellende Spannung [V]
curr	DBL	Einzustellende Strombegrenzung [A]
channel	INT	Kanal, den die Einstellungen betreffen sollen
success	BOOL	true, falls die Einstellungen übermittelt werden konnten

SASSetMode

Wählt den Modus des SAS-Ausgangs aus (FIX oder SAS).

Typ: Funktion
Klassifikation: SAS
Signatur: success = SASSetMode(mode<,channel>)

Ist channel angegeben, so wird nur der Modus für diesen Kanal gesetzt, ansonsten für beide.

Ein- und Ausgabeargumente

mode	STR	Zu aktivierender Modus ('FIX','SAS')
channel	INT	Kanal, den die Änderung betreffen soll (1 oder 2)
success	BOOL	true, falls die Operation ausgeführt werden konnte

SASSetSAS

Sendet die Einstellungen für den SAS-Modus an den SAS *ohne* den SAS-Modus zu aktivieren.

Typ:	Funktion
Klassifikation:	SAS
Signatur:	`success = SASSetSAS(Voc,Isc,Vmpp,Impp<,channel>)`

Ist `channel` angegeben, werden die Einstellungen nur an den entsprechenden Kanal gesendet, ansonsten an beide.

Ein- und Ausgabeargumente

`Voc`	DBL	Leerlaufspannung [V]
`Isc`	DBL	Kurzschlussstrom [A]
`Vmpp`	DBL	MPP-Spannung [V]
`Impp`	DBL	MPP-Strom [A]
`channel`	INT	Kanal, für den die Einstellungen gesetzt werden sollen (1 oder 2)
`success`	BOOL	`true`, falls die Einstellungen gesendet werden konnten

SASSetStatus

Setzt die handles der GUI `main`, die über den Zustand des SAS Aufschluss geben.

Typ:	Funktion
Klassifikation:	SAS
Signatur:	`SASSetStatus(handles)`

Eingabeargument

`handles`	HDL	handles aus der GUI `main`

SASSwitch

Schaltet den Ausgang / die Ausgänge des SAS ein oder aus.

Typ:	Funktion
Klassifikation:	SAS
Signatur:	`success = SASSwitch(action<,channel>)`

Ist `channel` angegeben, so wird nur dieser Kanal geschaltet, ansonsten beide.

Ein- und Ausgabeargumente

`action`	STR	’ON’ zum Einschalten, ’OFF’ zum Ausschalten
`channel`	INT	Kanal, der geschaltet werden soll (1 oder 2)
`success`	BOOL	`true`, falls die Schaltaktion ausgeführt werden konnte

SetHandles

Diese Funktion gestaltet die Benutzeroberfläche durch Aktivieren und Deaktivieren von Schaltflächen, das Ändern von Hintergrundfarben und Beschriftungen.

Typ: Funktion
Klassifikation: System
Signatur: `SetHandles(mode,handles)`

Eingabeargumente

`mode`	STR	Modus, für den die Handles gesetzt werden sollen ('BOOTED', 'SUSPENDED', 'LIVEVIEW', 'MEASUREMENT')
`handles`	HDL	Die Handles der GUI `main`

SetLiveViewVals

Trägt die Messwerte in den Schaltplan im *Arrangement Overview* ein.

Typ: Funktion
Klassifikation: System
Signatur: `SetLiveViewVals(arr,results,handles)`

Eingabeargumente

`arr`	CHR	Die Schaltungsart ('s' oder 'p')
`results`	STC	Ergebnisse, die eingetragen werden sollen[48]
`handles`	HDL	Handles der GUI `main`

SetMainSwitch

Setzt die Beschriftung und Funktionalität des Hauptschalters entsprechend dem Systemzustand.

Typ: Funktion
Klassifikation: System
Signatur: SetMainSwitch(handles)

Eingabeargument

`handles`	HDL	Handles der GUI `main`

SetMeasureHandlesVisibility

Schaltet die Schaltflächen im *Arrangment Overview* so sichtbar / unsichtbar, dass sie zur derzeitigen Schaltungsart passen.

Typ: Funktion
Klassifikation: System
Signatur: `SetMeasureHandlesVisibility(arr,handles)`

[48] Es werden folgende Felder mit DBL-Werten erwartet: v1.val, v2.val, v3.val, v4.val, v5.val, v6.val, v7.val, i1.val, i2.val, i3.val, i4.val, c1.ConvEff.val, c1.ConvEff.err, c1.MPPEff.val, c1.MPPEff.err, c1.TotEff.val, c1.TotEff.err, c2.ConvEff.val, c2.ConvEff.err, c2.MPPEff.val, c2.MPPEff.err, c2.TotEff.val, c2.TotEff.err

Eingabeargumente

`arr`	CHR	Spezifiziert die Schaltungsart ('s' oder 'p')
`handles`	HDL	Handles der GUI `main`

Start

Startet die Benutzeroberfläche und initialisiert nötige Variablen.

Typ: Skript
Klassifikation: START
Signatur: `Start`

SystemBoot

Startet das System, verbindet den PC mit den Messgeräten und nimmt die Grundeinstellungen (inklusive Laden der Standardwerte) vor.

Typ: Funktion
Klassifikation: System
Signatur: `SystemBoot(handles)`

Eingabeargument

`handles`	HDL	Handles der GUI `main`

SystemReset

Setzt alle verbundenen Geräte zurück.

Typ: Skript
Klassifikation: System
Signatur: `SystemReset`

SystemSuspend

Schaltet Last und SAS aus und beendet alle Verbindungen zu den Geräten.

Typ: Funktion
Klassifikation: System
Signatur: `SystemSuspend(handles)`

Eingabeargument

`handles`	HDL	Handles der GUI `main`

TMMDefaults

Gibt die Standardeinstellungen für die Multimeter zurück.

Typ: Funktion
Klassifikation: Einstellungen
Signatur: `[nplc,n,slope] = TMMDefaults`

Ausgabeargumente

`nplc`	DBL	Integrationszeit als NPLC
`n`	INT	Anzahl durchzuführender Messungen pro Triggersignal
`slope`	STR	Ausschlaggebende Flanke des Triggersignals ('pos' oder 'neg')

TMMErr

Berechnet die Messunsicherheit für eine gegebene Konstellation.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `error = TMMErr(id,read)`

Ein- und Ausgabeargumente

`id`	INT	Programminterne Identifikationsnummer des Multimeters
`read`	DBL	Gelesener Wert [V]
`error`	DBL	Messunsicherheit (einheitenfrei)

TMMErrorList

Gibt für eine bestimmte Messanordnung die Lese- und Bereichsunsicherheit des Multimeters einheitenfrei zurück.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `error = TMMErrorList(type,range,period)`

Ein- und Ausgabeargumente

`type`	STR	Gesuchter Fehlertyp ('read' oder 'range')
`range`	DBL	Messbereich (0, 0,1, 1, 10, 100, 1000) [V]
`period`	STR	Versuchsdauer ('24h', '90d' oder '1a')
`error`	DBL	Betreffende Unsicherheit (einheitenfrei)

TMMEvalAll

Wertet die Messungen aller Multimeter entsprechend des gegeben Aufbaus aus.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `results = TMMEvalAll(meas,arr)`

Ein- und Ausgabeargumente

`meas`	STC	Messwerte (durch `TMMReadAll` ausgelesen)
`arr`	CHR	Aktuelle Schaltungsart ('s' oder 'p')
`results`	STC	Alle Ergebnisse inklusive Messunsicherheiten (siehe Abschnitt A.3)

TMMInitAll

Bereitet alle Multimeter auf die nächste Messung vor.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `success = TMMInitAll(range,n,slope)`

Ein- und Ausgabeargumente

`range`	STR	Der zu verwendende Messbereich. 'auto' für AUTO-RANGE; 'get' für automatische Bestimmung; 'UseSettings', um aus den Systemparametern abzuleiten oder ein Wert aus {'100mV', '1V', '10V', '100V', '1000V'}
`n`	INT	Anzahl Messungen pro Triggersignal (1 bis 50 000)
`slope`	STR	Ausschlaggebende Flanke des Triggersignals ('pos' oder 'neg')

TMMIsValidParam

Prüft, ob ein gegebener Parameter für die Multimeter gültig ist.

Typ: Funktion
Klassifikation: Gültigkeit
Signatur: `valid = TMMIsValidParam(param,value)`

Ein- und Ausgabeargumente

`param`	STR	Parameter, für den ein Wert getestet werden soll ('nplc', 'numOfTrig')
`value`	DBL	Fraglicher Wert des Parameters
`valid`	BOOL	`true`, falls der Wert gültig ist, sonst `false`

TMMQuery

Sendet eine Anfrage an ein Multimeter und gibt die Antwort zurück.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `res = TMMQuery(query,id)`

Ein- und Ausgabeargumente

`query`	STR	Anfrage ('RANGE')
`id`	INT	Programminterne Identifikationsnummer des Multimeters
`res`	DBL	Antwort des Multimeters (Messbereich in [V])

TMMReadAll

Liest von allen Multimetern alle Werte aus.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `meas = TMMReadAll`

Ausgabeargument

`meas`	STC	Ergebnisse in der Form für `TMMEvalAll`

TMMSetAll2Wait

Versetzt alle Multimeter in den „Warte-auf-Trigger“-Zustand.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `success = TMMSetAll2Wait`

Ausgabeargument

`success`	BOOL	`true`, falls der Befehl für alle Multimeter ausgeführt wurde

TMMSetDefaults

Setzt für alle Multimeter die Standardeinstellungen.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `success = TMMSetDefaults`

Ausgabeargument

`success`	BOOL	`true`, falls für alle Multimeter die Standardeinstellungen geladen werden konnten.

TMMSort

Ordnet die Messergebnisse von interner Identifikationsnummer auf Bezeichnung der Größe im Schaltplan um.

Typ: Funktion
Klassifikation: Multimeter
Signatur: `sorted = TMMSort(var, field1, field2, ... , field7)`

Ein- und Ausgabeargumente

`var`	STC	Variable, die umsortiert werden soll
`field...`	STR	Bezeichnungen der Felder, die den Spannungen v1 ... v7 zugeordnet werden sollen (Reihenfolge!)
`sorted`	STC	Umsortiertes Struct mit neuen Feldnamen

TrigCheckParams

Prüft die Gültigkeit von Triggereinstellungen.

Typ: Funktion
Klassifikation: Gültigkeit
Signatur: `[ok,mess] = TrigCheckParams(<n,v,pw>)`

Sind `n`, `v` und `pw` angegeben, so werden diese Werte geprüft; anderenfalls die Werte der globalen Variablen `state`.

Ein- und Ausgabeargumente

`n`	INT	Anzahl Messungen
`v`	DBL	Spannung des Triggersignals [V]
`pw`	DBL	Pulsbreite des Triggersignals [s]
`ok`	BOOL	`true`, falls die Werte gültig sind, sonst `false`
`mess`	STR	Meldung, welcher Wert nicht gültig ist

TrigDefaults

Gibt die Standardeinstellungen für den Trigger zurück.

Typ: Funktion
Klassifikation: Einstellungen
Signatur: `[n,nplc,volt,pw] = TrigDefaults(type)`

Ein- und Ausgabeargumente

`type`	STR	Unterscheidung verschiedener Standardwerte ('default', 'live', 'test')
`n`	INT	Anzahl Messungen pro Triggersignal
`nplc`	DBL	Integrationszeit in NPLC
`volt`	DBL	Spannung des Triggersignals [V]
`pw`	DBL	Pulsbreite des Triggersignal [s]

TrigExec

Sendet den Befehl, den Trigger auszulösen.

Typ: Skript
Klassifikation: Trigger
Signatur: `TrigExec`

TrigLimits

Gibt die Begrenzungen für den Trigger zurück.

Typ: Funktion
Klassifikation: Einstellungen
Signatur: `[...] = TrigLimits`

Die folgenden Werte werden in dieser Reihenfolge (links oben nach rechts unten) zurückgegeben:

INT	Mindestanzahl Messungen pro Trigger	DBL	Minimale Triggerspannung [V]
INT	Maximalanzahl Messungen pro Trigger	DBL	Maximale Triggerspannung [V]
DBL	Mindestfrequenz des Triggersignals [kHz]	DBL	Minimale Pulsbreite [s]
DBL	Maximalfrequenz des Triggersignals [kHz]	DBL	Maximale Pulsbreite [s]

TrigSet

Sendet Einstellungen an den Trigger.

Typ: Funktion
Klassifikation: Trigger
Signatur: `success = TrigSet(n,nplc,volt,pw)`

Ein- und Ausgabeargumente

`n`	INT	Anzahl Messungen pro Triggersignal
`nplc`	DBL	Integrationszeit in NPLC
`volt`	DBL	Spannung des Triggersignals [V]
`pw`	DBL	Pulsbreite des Triggersignals [s]
`success`	BOOL	`true`, falls die Einstellungen übermittelt wurden

TrigSetStatus

Setzt die Anzeigen in der GUI `main` für die Triggerwerte.

Typ: Funktion
Klassifikation: Trigger
Signatur: `TrigSetStatus(handles)`

Eingabeargument

`handles`	HDL	Handles der GUI `main`

VarDevices

Initialisiert die globalen Variablen `devices` und `ind`.

Typ: Skript
Klassifikation: Einstellungen
Signatur: `VarDevices`

VarState

Setzt zum Starten des Systems nötige Standardwerte für einige Felder der globalen Variablen `state`.

Typ: Skript
Klassifikation: Einstellungen
Signatur: `VarState`

B Vorgehen beim Austausch eines Gerätes

Soll ein Teil des Teststandes ausgetauscht werden, so sind einige Dinge zu beachten:

Jedes der (aktiven) Geräte hat eine *programminterne Identifikationsnummer* und (meistens) eine *zugewiesene physikalische Größe*. Um eine saubere Auswertung sicherzustellen, muss also darauf geachtet werden, dass Identifikationsnummer und pysikalische Größe einander korrekt zugeordnet werden. Die Zuordnung, wie sie derzeit besteht, ist in Tabelle 3.1 gegeben. Es wird empfohlen, die Geräte stets wie in dieser Tabelle angegeben zuzuordnen.

Ist es nötig, ein aktives Gerät auszutauschen, so müssen einige Anpassungen vorgenommen werden: Im Skript `VarDevices` wird jeder Identifikationsnummer eine Seriennummer (und einige andere Eigenschaften) zugeordnet. Diese Eigenschaften müssen bei einem Gerätetausch entsprechend modifiziert werden. Wird auch noch die Identifikationsnummer geändert – wovon *dringend* abgeraten wird – so muss auch die Funktion `TMMSort` angepasst werden.

Für die Widerstände sind verschiedene Parameter – wie z. B. der exakte Wert – in der Funktion `AskParam` abgelegt. Wird einer der Widerstände ausgetauscht, so müssen dort die entsprechenden Werte angepasst werden. Werden die Werte nicht angepasst, sind sämtliche Berechnungen nicht mehr so exakt wie möglich.

C Technische Daten der Messgeräte

C.1 Solar Array Simulator

Modus	*Größe*	*Wert*
SAS	Maximale Leistung	498 W
	Maximale Leerlaufspannung $V_{\text{oc,max}}$	58 V
	Maximale MPP-Spannung $V_{\text{mpp,max}}$	53,5 V
	Maximaler Kurzschlussstrom $I_{\text{sc,max}}$	9,3 A
	Maximaler MPP-Strom $I_{\text{mpp,max}}$	9,3 A
	Kleinste Impedanz dV/dI für $V > V_{\text{mpp}}$ [49]	0,25 Ω
FIX	Ausgangsspannungsbereich	0 - 53,5 V
	Ausgangsstrombereich	0 - 9,3 A

Tabelle C.1: *Technische Daten der SAS-Module des Typs E4361A-J01 bei einer Netz-Nennspannung von 230* V∼. *Nach [Agilent SAS DB, p. 9]*

C.2 Tischmultimeter

Kenngröße	*Wert*
Eingangsimpedanz	10 MΩ ± 1%
Messbereiche	100 mV, 1 V, 10 V, 100 V, 1000 V
Messmethode	Kontinuierlich intergrierender Multi-Flanken IV A/D-Wandler
Eingangs-Verzerrungsstrom	< 50 pA bei 20°C

Tabelle C.2: *Daten der Agilent 34410A-Multimeter für Gleichspannungsmessungen. [Agilent TMM DB, p. 5]*

NPLC	*Messber.bez. Auflösung*	*Absolute Auflösung*	*Messungen / Sek. (max.)*
0,006	6 ppm	$\Delta V \leq 6$ mV	10 000
0,02	3 ppm	$\Delta V \leq 3$ mV	3 000
0,06	1,5 ppm	$\Delta V \leq 1{,}5$ mV	1 000
0,2	0,7 ppm	$\Delta V \leq 700\ \mu$V	300
1	0,3 ppm	$\Delta V \leq 300\ \mu$V	50
2	0,2 ppm	$\Delta V \leq 200\ \mu$V	25
10	0,1 ppm	$\Delta V \leq 100\ \mu$V	5
100	0,03 ppm	$\Delta V \leq 30\ \mu$V	0,5

Tabelle C.3: *Auflösungen und Messgeschwindigkeiten in Abhängigkeit der NPLC für DC-Messungen. Nach [Agilent TMM DB, p. 4]*

[49] Laut [Agilent SAS DB] beträgt die kleinste Impedanz 1 Ω; gemäß Rückmeldungen des Gerätes auf SCPI-Befehle jedoch die hier angegebenen 0,25 Ω.

Messbereich	*Zeitfenster*[50]		
	24 h	*90 Tage*	*1 Jahr*
100 mV	0,0030 + 0,0030	0,0040 + 0,0035	0,0050 + 0,0035
1 V	0,0020 + 0,0006	0,0030 + 0,0007	0,0035 + 0,0007
10 V	0,0015 + 0,0004	0,0020 + 0,0005	0,0030 + 0,0005
100 V	0,0020 + 0,0006	0,0035 + 0,0006	0,0040 + 0,0006
1000 V	0,0020 + 0,0006	0,0035 + 0,0006	0,0040 + 0,0006

Tabelle C.4: *Unsicherheiten der Messungen für die Tischmultimeter, jeweils als % des Messwerts + % der Bereichsobergrenze. Nach [Agilent TMM DB, p.3]*

C.3 burster Hochpräzisionswiderstand

Widerstand (bez. auf Abb. 3.2)	*Inv.-Nr.*	*Kalibrations-Zert.-Nr.*	*Nominalwert*	*Realwert*	*Ungenauigkeit*	*Therm. Widerstand*	$\overline{\alpha}$
R_1	W0428	7320	10 mΩ	10,00161 mΩ	0,02%	1 K/W	0,001%/K
R_2	W0429	7321		10,00128 mΩ			
R_3	W0430	7322		10,00122 mΩ			
R_4	W0427	5162		10,00040 mΩ			

Tabelle C.5: *Daten der Hochpräzisionswiderstände nach [Burster o.J., p. 2]*

C.4 Elektronische Last

Modus	*Parameter*	*Wert*
allg.	Leistung	2600 W
	Laststrom	0 A bis 50 A
	Lastspannung	1,25 V bis 500 V
CC	Bereich	0 A bis 50 A
	Auflösung	12,5 mA
CR	Bereich	10 Ω bis 40 kΩ
	Auflösung	12 bits
CV	Bereich	2,5 V bis 500 V
	Auflösung	125 mV
CP	Bereich	6,25 W bis 2,6 kW
	Auflösung	31,25 mW

Tabelle C.6: *Wichtigste Daten der elektronischen Last nach [Chroma HB, p. 1-4]*

[50] Das Zeitfenster gibt an, in welchen Zeitraum alle Messungen liegen müssen, um dieser Genauigkeit zu entsprechen.



Eigenständigkeitserklärung

Ist jeder an der ETH verfassten schriftlichen Arbeit unterzeichnet beizufügen.

Ich erkläre hiermit, dass es sich bei der von mir eingereichten schriftlichen Arbeit mit dem Titel

Automatischer Teststand für solarmodulintegrierte DC-DC-Konverter

um eine von mir selbständig und in eigenen Worten verfasste Originalarbeit handelt.

VerfasserIn/VerfasserInnen

Name	Vorname
Bomhauer-Beins	Axel

Betreuende/r DozentIn

Name	Vorname
Kolar	Prof. Dr. Johann W.

Mit meiner Unterschrift bestätige ich, dass ich über fachübliche Zitierregeln unterrichtet worden bin und das Merkblatt (http://www.ethz.ch/students/exams/plagiarism_s_de.pdf) gelesen und verstanden habe. Die im betroffenen Fachgebiet üblichen Zitiervorschriften sind eingehalten worden.

Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden

Ort, Datum

Unterschrift

* Bei Gruppenarbeiten sind die Unterschriften aller VerfasserInnen erforderlich.
Durch die Unterschrift bürgen Sie für den vollumfänglichen Inhalt der Endversion dieser schriftlichen Arbeit.

Formular drucken